

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего  
образования  
«Тольяттинский государственный университет»

Кафедра \_\_\_\_\_ «Прикладная математика и информатика»  
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки)

Разработка социальных и экономических информационных систем

(направленность(профиль))

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
(БАКАЛАВРСКАЯ РАБОТА)**

На тему \_\_\_\_\_  
Разработка системы управления

\_\_\_\_\_ многоквартирным домом для ТСН/ТСЖ

Обучающийся \_\_\_\_\_  
Хенкин Г.Р.

\_\_\_\_\_ (Инициалы Фамилия)

\_\_\_\_\_ (личная подпись)

Руководитель \_\_\_\_\_  
к.п.н., доцент Е.А. Ерофеева

\_\_\_\_\_ (ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант \_\_\_\_\_  
к.п.н., доцент И.Ю. Усатова

\_\_\_\_\_ (ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

## Аннотация

Выпускная квалификационная работа выполнена по теме «Разработка информационной системы для управления многоквартирным домом ТСН/ТСЖ».

Объектом исследования является процесс управления многоквартирным домом. Предметом исследования является автоматизация процесса управления многоквартирным домом. Актуальность темы обусловлена потребностью председателей ТСН в упрощении и автоматизации процессов, проводимых ими при самостоятельном управлении многоквартирными домами.

Цель работы – разработка информационной системы управления многоквартирным домом. В работе рассмотрены проблемы, возникающие при самостоятельном управлении многоквартирным домом его собственниками, а также способы их решения путём внедрения разрабатываемой информационной системой. В результате работы достигнута цель и решены поставленные задачи, результатом является веб-приложение, готовое для использования председателями ТСН и собственниками квартир.

В работе использованы актуальные методы проектирования информационных систем: Диаграммы классов, последовательностей. Реализована модульная архитектура приложения, состоящая из модулей: хранения, обработки, представления данных, а также бота для оперативной связи с пользователями.

Разработана система для автоматизации и упрощения взаимодействия правления ТСН с остальными его участниками, помогающая в планировании и реализации мероприятий по содержанию общедомовой собственности. Решена проблема длительного ожидания ответов на обращения, недостаточной осведомлённости жильцов, сложности принятия решений по распределению бюджета и времени на проведение необходимых для общего имущества мероприятий.

Выпускная квалификационная работа состоит из 50 страниц.

## **Abstract**

The title of the graduation work is “The Development of an Information System for Managing a Homeowner Association (TSG/TSH)”.

The graduation project deals with the complexity of multi-apartment building management and automation to enhance and optimize the responsibilities handled by TSG chairpersons. The topic's significance is underscored by an urgent demand for simplified and automated solutions that are expected to elevate the self-management practices of multi-apartment properties by their proprietors.

The graduation work consists of an explanatory note on fifty pages, an introduction, including thirty six figures, the list of twenty five references including five foreign sources.

We touch upon the problem of automating the duties of the TSG chairperson.

The aim of the work is the development of an information system for managing a multi-apartment building.

The object of the graduation work is the interaction of the Homeowner Association Board with its members, resource supplying organizations, and external contractors for repair work.

The subject of the graduation work is the automation of the process of managing a multi-apartment building by the Homeowner Association.

The graduation work may be divided into several logically connected parts. The first part considers the complexity of managing properties in a multi-apartment context. Then the logical modeling of the proposed system is done, offering a thorough layout for its anticipated design and operational capabilities. In the third part, the attention is given to the actual assembly and evaluative testing of the system, with the emphasis on achieving operational efficacy, user accessibility, and dependability for all intended users.

The author comes to the conclusion that the information system developed in accordance with the set objectives has a significant positive impact on the process of managing a multi-apartment building.

## Оглавление

Введение .....	5
Глава 1 Характеристика организации и анализ существующих решений .....	7
1.1 Теоретические основы Товариществ собственников недвижимости.....	7
1.2 Общая характеристика предприятия.....	10
1.3 Моделирование бизнес-процессов .....	15
1.3.1 Модель бизнес-процесса «AS-IS» .....	17
1.3.2 Модель бизнес-процесса «TO BE».....	18
1.4 Анализ существующих разработок.....	19
Глава 2 Логическое проектирование .....	22
2.1 Логическая модель информационной системы .....	22
2.2 Разработка логической модели данных .....	41
Глава 3. Тестирование информационной системы .....	43
3.1 Модульное тестирование .....	43
3.2 Функциональное тестирование.....	54
Заключение .....	59
Список используемой литературы.....	60

## Введение

На данный момент в сфере ЖКХ часто используются устаревшие способы организации многих процессов, в том числе и в области управления многоквартирными домами. Многие собственники несут избыточные финансовые расходы из-за пользования услугами управляющих компаний, потому что не готовы к самостоятельному планированию общедомового бюджета и решению различных проблем, порождаемых отсутствием автоматизации процессов. В связи с этим возникает потребность во внедрении информационной системы, которая позволит автоматизировать процесс планирования работ по ремонту и содержанию дома, а также оптимизировать расходы жильцов, входящих в ТСН.

Для решения этой проблемы необходимо разработать приложение, которое будет автоматически рассчитывать расходы бюджета, своевременно уведомлять участников о важных событиях, систематизирует все документы, необходимые для управления многоквартирным домом, будет оперативно предоставлять председателю доступ к ним в электронном виде.

Объектом исследования является взаимодействие правления ТСН с его участниками, ресурсоснабжающими организациями и внешними исполнителями ремонтных работ.

Предметом исследования является автоматизация процесса управления многоквартирным домом ТСН.

Целью выпускной квалификационной работы является разработка информационной системы управления многоквартирным домом.

Для достижения цели требуется решить следующие задачи:

- изучить законодательство по теме управления многоквартирными домами, требования к ТСН
- изучить текущие методы организации взаимодействия между жильцами и управляющими организациями
- проанализировать существующие программные решения в рамках предметной области

- выбрать инструменты и технологии программной реализации информационной системы
- произвести проектирование системы и разработать её структуру
- реализовать и протестировать систему

Научная новизна работы заключается в разработке системы управления многоквартирным домом, которая позволяет правлению дома снизить бюрократическую нагрузку, спланировать мероприятия по содержанию дома и наладить коммуникацию с собственниками жилья с целью оптимизации общего бюджета.

Работа состоит из трёх глав. В первой главе проводится анализ и сравнение текущих решений, даётся их характеристика, обосновываются причины отказа от них. Во второй главе описываются требования к разрабатываемому приложению, описывается процесс разработки диаграмм классов, а также взаимосвязи и последовательности процессов взаимодействия разных участников системы. После этого осуществляется обоснование выбора инструментов реализации. В третьей главе описываются инструменты разработки и процесс разработки.

# **Глава 1 Характеристика организации и анализ существующих решений**

## **1.1 Теоретические основы Товариществ собственников недвижимости**

Товарищество собственников недвижимости - это объединение людей, владеющих квартирами, офисными помещениями, землей, дачными домами и участками, служебными постройками.

Цель объединения владельцев – организация мероприятий по благоустройству, сохранению в хорошем состоянии общей собственности. По закону участники товарищества имеют право регулировать вопросы, связанные с использованием и поддержанием исправного состояния самих зданий и коммуникаций.

Задачи и порядок их исполнения фиксируются в уставе товарищества собственников недвижимости.

Участники – владельцы недвижимого имущества – могут входить в состав товарищества. Это имущество должно представлять некое единство – входить в один поселок, жилой комплекс, находиться внутри одного строения, обслуживаться одними коммуникациями, располагаться в соседних постройках. Земельные наделы должны составлять единый массив. Управлять объектами, объединенными произвольно, невозможно.

Для регистрации такого объединения необходимо провести собрание жильцов и провести голосование. Если инициатива поддержана более чем половиной голосов, регистрация считается возможной.

Основной учредительный документ объединения – Устав ТСН. Он необходим для регистрации товарищества в налоговой службе. Содержит сведения о должностных лицах, положения по организации управления общим имуществом, контроля за исполнением обязанностей. Титульный лист Устава содержит:

- наименования документа;
- организационно-правовой формы объединения;

- названия товарищества;
- города и года создания документа;
- ФИО и подписи председателя, утвердившего устав.

Документ оформляется не только в электронном, но и в бумажном виде.

Распечатанный устав всегда должен храниться у председателя и правления.

Устав должен содержать следующие основные пункты:

- название (полное и сокращенное), которое включает обозначение правовой формы «товарищество собственников недвижимости»
- местонахождение – название города, поселка, деревни
- предмет и цели создания – совместное использование общей собственности
- порядок создания
- структура – правление, председатель
- состав и полномочия органов правления, пути осуществления полномочий
- процедура и порядок принятия решений

Также в Уставе могут быть прописаны:

- права и обязанности владельцев недвижимости;
- правила принятия и исключения из членов товарищества;
- порядок расчета и уплаты членских взносов;
- правила созыва и проведения общего собрания;
- последовательность действий при заключении договоров с подрядчиками, сторонними организациями;
- правила использования общего имущества;
- порядок ведения бухгалтерского учета;
- последовательность действий в случае реорганизации и ликвидации товарищества.

Перед получением Уставом юридической силы, необходимо соблюсти ряд условий:



- Все владельцы недвижимости должны быть проинформированы о предстоящем собрании, знать точные дату, время и место его проведения. Допустимо провести информирование через электронную почту или мессенджеры, а также путем размещения объявления на информационной доске товарищества.

- Для утверждения устава требуется обеспечить кворум собственников. В случае его наличия, документ считается утвержденным при условии, что по результатам голосования большинство участников собрания одобрили инициативу.

- Если не удалось обеспечить появление необходимого числа жителей, возможно провести голосование заочно после ознакомления с уставом.

- Устав считается утвержденным, при получении поддержки не менее двух третей собственников.

- После утверждения, устав прошивается, и председатель ставит на нем свою подпись. Подписи других членов товарищества на документе не требуются.

При завершении подготовки документа следует начать процедуру регистрации Товарищества Собственников Недвижимости (ТСН) в налоговом органе по месту его расположения. Устав, подписанный председателем, прошитый и пронумерованный, представляется в Федеральную налоговую службу в двух экземплярах вместе с комплектом других документов. После процесса регистрации ТСН, устав приобретает статус полноценного и фундаментального документа данного объединения.

Устав ТСН может быть изменён после утверждения. Обычно этим занимается председатель товарищества, для этого ему необходимо:

- Организовать собрание собственников недвижимости;
- Официально обозначить предложенные поправки;
- Провести голосование по каждому предложенному пункту;
- Утвердить решение общего собрания и формирование соответствующего протокола.

Инициатива по корректировке Устава может исходить не только от председателя или правления товарищества, но и от его членов. Все предложения должны быть представлены председателю в письменном виде.

Для осуществления поправок в Уставе, необходимо получить поддержку не менее двух третей собственников недвижимости, входящих в состав товарищества. После внесения всех изменений и дополнений, обновленный Устав подлежит повторной регистрации в налоговом органе. Все поправки вступают в законную силу только после этого.

## 1.2 Общая характеристика предприятия

Компания ООО «Квартплата 24» оказывает услуги в сфере жилищно-коммунального хозяйства для управляющих компаний, ресурсоснабжающих организаций и товариществ собственников недвижимости. Она предоставляет услуги по расчёту и учёту платы за ЖКХ, централизованный приём и распределение платежей между организациями-получателями. Также клиенты могут получить доступ к сервисам: аналитики, автоматического информирования жителей, взыскания долгов и т.д.

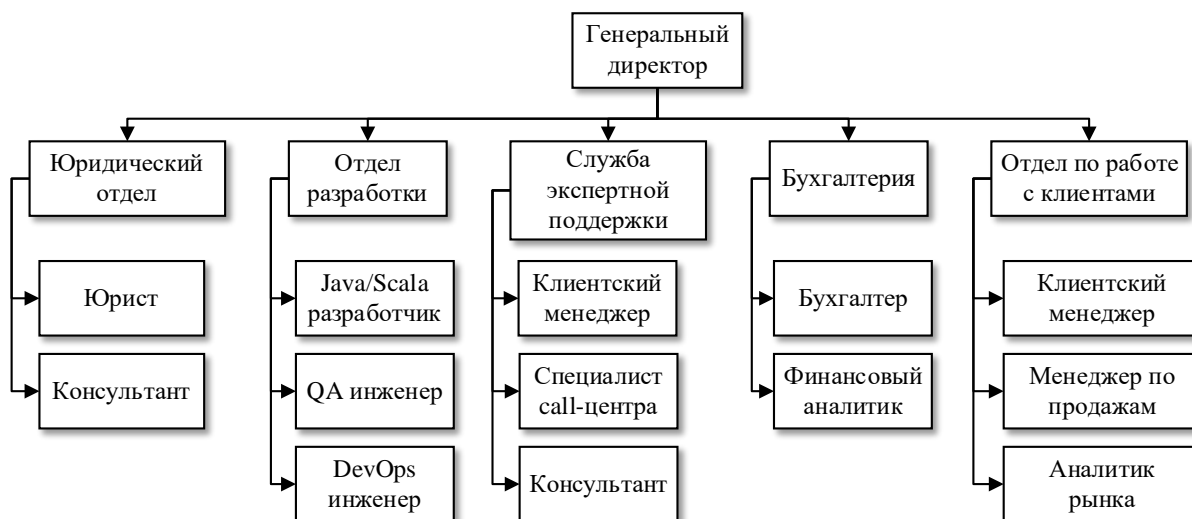


Рисунок 1 – Организационная диаграмма ООО «Квартплата 24»

Работу компании обеспечивают 5 отделов, каждый из которых отвечает за свои задачи. Все они подчиняются генеральному директору. Структура компании отображена на рисунке 1.

Юридический отдел разрабатывает и редактирует договоры, контракты, соглашения, контролирует соответствие деятельности компании действующим законам и требованиям. Так как ООО «Квартплата 24» предоставляет услуги в сфере ЖКХ, то отслеживание изменений в нормативных документах, таких как «Постановление Правительства РФ "О предоставлении коммунальных услуг собственникам и пользователям помещений в многоквартирных домах и жилых домов"» и своевременное консультирование отдела разработки при появлении правок, напрямую влияет на корректность расчётов в сервисах, предоставляемых компанией. Без юридического отдела у компании есть риск репутационного ущерба и административной ответственности в случае проведения неверных расчётов стоимости жилищно-коммунальных услуг.

Бухгалтерия управляет финансовыми операциями компании и ведёт финансовую отчётность. В обязанности отдела входит разработка финансовых планов и бюджетов, прогнозирование доходов и расходов, оценка финансового состояния компании и предложение рекомендаций по улучшению финансовых показателей. Особую важность представляет ведение финансовой и налоговой отчётности. Во избежание ответственности, компании необходимо правильно рассчитывать налоги, своевременно их уплачивать и предоставлять налоговую отчётность в Федеральную налоговую службу. Такая отчётность должна готовиться ежеквартально, поэтому с течением времени бухгалтерия не теряет своей актуальности.

Служба экспертной поддержки призвана помочь пользователям сервисов компании справиться с трудностями при использовании сервисов, в случае возникновения таковых. При использовании продуктов, разработанных компанией ООО «Квартплата 24», по различным причинам могут возникать непредвиденные сложности, на которые необходимо реагировать и помогать

клиентам, для исполнения своих обязательств и сохранения лояльности клиентов. Пользователи могут обращаться к справочной информации, но её может оказаться недостаточно, так как некоторые проблемы возникают из-за технических сбоев и ошибок в программном обеспечении. На этот случай предусмотрен отдел, который принимает обращения клиентов и даёт им консультацию по возникшим проблемам. Если проблему невозможно решить без помощи разработчиков, сотрудники отдела поддержки уточняют подробности возникшей неисправности и передают их в отдел разработки.

Отдел по работе с клиентами занимается поиском и привлечением новых клиентов. Так как компания уже имеет определённую клиентскую базу и отлаженные бизнес-процессы, она может расширяться, увеличивая число пользователей своих услуг. Расширение круга пользователей сервисов компании позволяет ей получить некоторые преимущества: рост доходов, устойчивость бизнеса, оптимизация ресурсов, увеличение узнаваемости бренда. Подходы к увеличению числа клиентов различаются в зависимости от выбранной руководством компании стратегии масштабирования. Различают горизонтальное и вертикальное масштабирование.

Горизонтальное масштабирование для ООО «Квартплата 24» означает расширение базы пользователей существующих сервисов. Это включает в себя привлечение новых клиентов, которые будут использовать те же услуги, что уже предлагает компания. Горизонтальное масштабирование касается увеличения объема продаж и расширения рыночного охвата без изменения сущности предлагаемых услуг.

Вертикальное масштабирование включает в себя внедрение и развитие новых услуг или продуктов. Вертикальное масштабирование предполагает расширение спектра услуг компании и обращение к клиентам, которые могут быть заинтересованы в этих новых видах услуг или продуктах.

Обычно отдел по работе с клиентами занимается поиском и привлечением пользователей для существующих услуг. В результате анализа потребности текущих клиентов и существующих решений от других

компаний, отделом была выдвинута гипотеза о возможности вертикального масштабирования, путём предоставления новых услуг уже существующим клиентам среди товариществ собственников жилья. Компания ООО «Квартплата 24» оказывает услуги в сфере ЖКХ более чем 500 товариществам собственников жилья, что увеличивает шансы на обретение популярности потенциального сервиса, упрощая поиск заинтересованных в нём организаций. Таким образом наибольшей сложностью для вертикального масштабирования компании является разработка такого сервиса.

Отдел разработки программного обеспечения. На начальном этапе работы компании занимался созданием сервисов в сфере услуг ЖКХ, на данный момент занимается их поддержкой и доработкой. Разработчики следят за стабильностью работы всех систем, обеспечивают их надёжность, своевременно реагируют на сбои и устраняют неполадки, информацию о которых получают, обычно от службы экспертной поддержки. Для компании отдел разработки критически важен, так как обеспечение стабильности работы сервисов и своевременное исправление ошибок влияет на качество предоставляемых клиентам услуг.

Для предоставления клиентам услуг используется экосистема облачных сервисов. Все они представляют собой готовое решение для расчета платы за жилищно-коммунальные услуги, формирование платежных документов, распределения платежей, работы с дебиторской задолженностью, обмена данными с ГИС ЖКХ, проведения аналитики и управления внутренними процессами организации.

Сервисы, разработанные для клиентов отображены на рисунке 2. В их числе имеются следующие:

- Биллинговая система – автоматически рассчитывает платы за услуги ЖКХ, также отвечает за распределение платежей между организациями-получателями (PCO и CO) без участия управляющих организаций и товариществ собственников недвижимости.

- Личный кабинет жителя – предназначен для взаимодействия плательщиков с экосистемой компании. Предоставляет удобный интерфейс для внесения показаний счётчиков, оплаты счетов, просмотра платёжных документов, своевременного получения информации о начислениях за ЖКУ и другие возможности.
- сервис аналитики – используется руководителями обслуживающих организаций, и предоставляет им информацию о начислениях, поступлении и расщеплении платежей по управляемому жилому фонду;
- сервис работы с должниками (он же СРД) – предназначен для проведения досудебной и судебной работы с должниками клиентов.

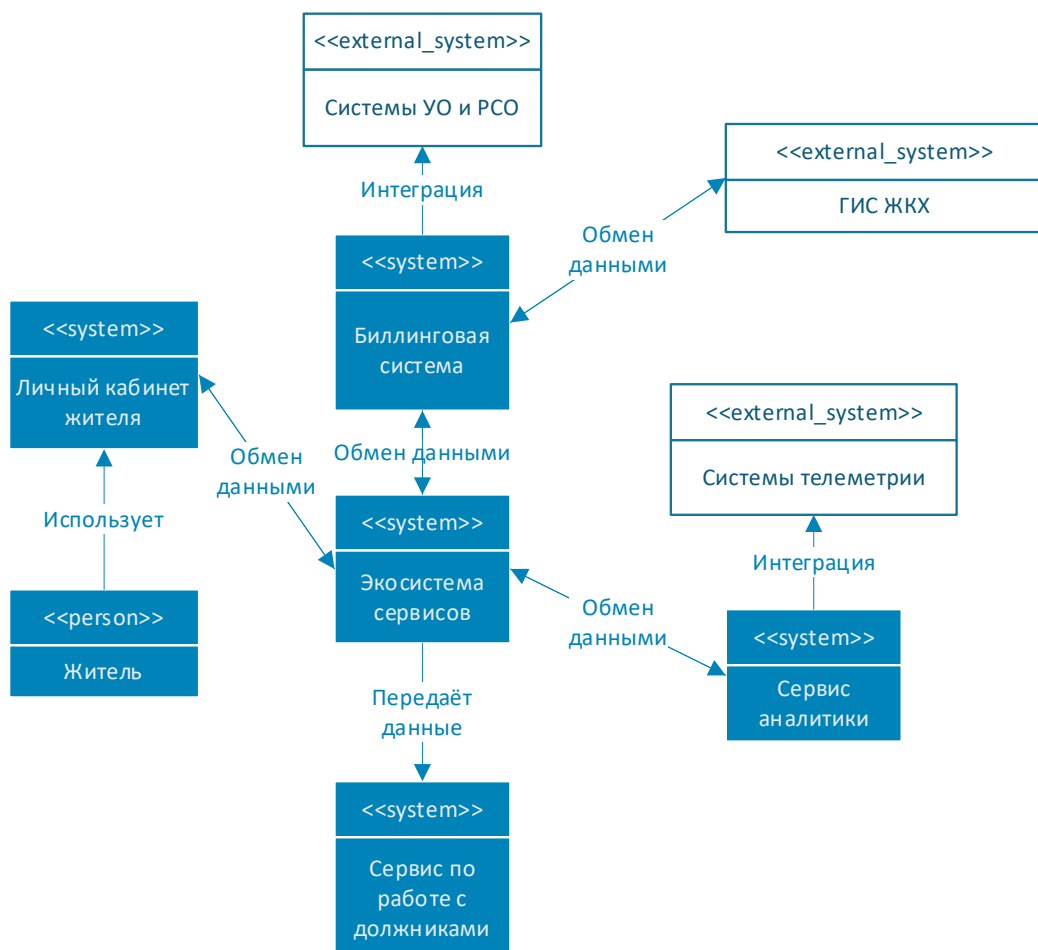


Рисунок 2 – Диаграмма потоков данных между сервисами в ООО  
«Квартплата 24»

Сервисы экосистемы, разработанной ООО «Квартплата 24» закрывают определённые потребности клиентов, но среди них нет комплексного решения для жильцов, желающих перейти на самостоятельное управление домом. Из-за отсутствия упомянутого сервиса, ООО «Квартплата 24» упускает возможность привлечь новых клиентов, за счёт расширения спектра услуг в сфере ЖКХ.

На текущий момент использование автоматизированных информационных систем среди товариществ собственников недвижимости непопулярно из-за дороговизны и сложности внедрения. Существующие решения подразумевают полный переход на собственную экосистему, которая может оказаться непривычной для потенциальных пользователей, в то время как ООО «Квартплата 24» имеет собственную базу клиентов, среди которых можно открыть новую категорию пользователей экосистемы.

### **1.3 Моделирование бизнес-процессов**

Процесс — это устойчивая, целенаправленная совокупность взаимосвязанных видов деятельности, которая по определенной технологии преобразует входы в выходы, представляющие ценность для потребителя.

Бизнес-процесс является особым процессом, который служит осуществлению основных целей предприятия или организации (бизнес-целей) и описывает центральную сферу его деятельности.

Моделирование бизнес-процессов – это набор действий, создающих представление существующего или предполагаемого бизнес-процесса.

Существует несколько подходов к моделированию бизнес-процессов:

- IDEF (Integrated DEFinition Methods):

- IDEF0 – Используется для моделирования функций бизнеса, где сложные функции делятся на более мелкие, взаимосвязанные части.
- IDEF3 – Фокусируется на описании последовательности операций и взаимодействия бизнес-процессов.
- BPMN (Business Process Model and Notation) – Стандартизированный подход для моделирования бизнес-процессов, который позволяет создавать детальные диаграммы, понятные как бизнес-аналитикам, так и техническим специалистам.
- UML (Unified Modeling Language) – Язык моделирования, применяемый в основном для разработки программного обеспечения, но также используемый для бизнес-моделирования.
- EPC (Event-driven Process Chain) – Метод, ориентированный на события, который включает в себя события, функции и соединительные элементы, такие как логические операторы.

В данной работе рассматривается потенциал использования разрабатываемой информационной системы, которая управляет процессами, проходящими в организации клиента. Так как клиентами являются товарищества собственников недвижимости, то необходимо рассмотреть процессы взаимодействия собственников квартир, правления товарищества и внешних специалистов. Кроме того, необходимо рассмотреть использование существующих сервисов ООО «Квартплата24» в рамках создаваемой системы.

Так как ТСН не имеет сложной структуры, не содержит большого числа отделов, а акцент при моделировании важнее сделать на последовательности процессов и взаимодействиях между их участниками, то наиболее подходящей выглядит нотация BPMN.

Основными компонентами диаграмм данной нотации являются события и действия (задания), которые объединены потоками управления. Под потоком управления понимается определенная последовательность этапов исполнения бизнес-процесса. Событием обозначается конкретный



наступивший факт либо инцидент, возникающий в ходе осуществления процесса или как итог выполнения определенных действий. Процесс на BPMN-диаграмме инициируется событием и завершается событием.

### **1.3.1 Модель бизнес-процесса «AS-IS»**

Участниками Товарищества собственников жилья (ТСЖ), являются: председатель, собрание собственников, ресурсоснабжающая организация (РСО) и внешние специалисты, задействованные для осмотра и ремонта дома, процесс может быть описан следующим образом:

Председатель ТСЖ организует и проводит собрание собственников для обсуждения текущих вопросов управления домом, включая необходимость проведения осмотра или ремонта. Собрание проводится очно, извещение отправляется в письменном виде всем участникам собрания, также вывешивается анонс собрания на информационном стенд дома.

На очном собрании собственники голосуют и принимают решение о необходимости проведения технического осмотра или ремонта дома, обсуждается и утверждается объем предстоящих работ. Проводится ручной подсчёт голосов, результаты голосования фиксируются в годовом плане работы ТСН.

Председатель на основании решений собрания вступает в переговоры и заключает договоры с РСО и внешними специалистами (инженерами, ремонтными бригадами и прочими подрядчиками). Заключенные договоры сохраняются в архиве ТСН, жильцам предоставляется доступ к просмотру этих документов.

Внешние специалисты проводят технический осмотр дома для оценки его состояния и определения объема нужных работ. Отчёты об их работе ведутся в бумажном формате и хранятся в архиве ТСН.

По результатам осмотра специалисты готовят технический отчет, который передается в ТСЖ для ознакомления собственников и определения

дальнейших действий. Для обсуждения требуется организовывать новое собрание собственников.

После принятия решения о ремонте председатель ТСЖ координирует и контролирует ремонтные работы, обеспечивая взаимодействие между РСО, внешними специалистами и собственниками. Отчёты о выполненных работах и использованных средствах хранятся в архиве ТСН.

В течение всего процесса председатель ТСЖ также занимается оформлением необходимой документации, включая акты выполненных работ, договора на поставку материалов, платежные документы и прочее. Эти документы доступны собственникам дома, но доступны только в бумажной версии, что приводит к дополнительным сложностям при желании ознакомиться с ними.

### **1.3.2 Модель бизнес-процесса «ТО ВЕ»**

Внедрение информационной системы изменит устоявшиеся в ТСЖ процессы. Прежде всего, это коснётся коммуникаций и обработки информации: организация собраний трансформируется благодаря системному распространению приглашений и мгновенным уведомлениям пользователей через функцию push-сообщений. Перевод очных голосований на дистанционные, с применением автоматизированной системы сбора и подсчёта голосов, повышается скорость получения результата и прозрачность процедур подсчёта.

Автоматизация расчетов и финансового учета, включая аспекты расходов и взносов собственников, исключает необходимость ручного контроля над этими операциями. Такая система не только минимизирует вероятность ошибок, но и позволяет более грамотно планировать обслуживание дома, в том числе необходимые ремонтные работы, предоставляя доступ к актуальной информации о ходе запланированных мероприятий.

Централизованный электронный документооборот, позволяющий жильцам получать доступ к документам в соответствии со своей ролью в управлении многоквартирным домом, а также видеть актуальные отчёты о ходе ремонтных работ, позволит оперативно реагировать на возможные срывы сроков завершения мероприятий по обслуживанию или ремонту дома.

Информационная система также станет фундаментом для оптимизации внутренней коммуникационной среды ТСН, создавая унифицированную платформу для эффективного общения собственников с правлением и между собой, что будет способствовать повышению оперативности реагирования на непредвиденные ситуации и повышению вовлечённости жильцов в обсуждение актуальных проблем многоквартирного дома.

#### **1.4 Анализ существующих разработок**

Информационные системы, подобные данной уже существуют на рынке. Перед проектированием собственной ИС стоит рассмотреть возможности конкурентов. Одним из них является платформа Doma.ai.

Сервисы от компании "Doma.ai" включают следующие возможности:

- Диспетчерская служба: Автоматизация приема заявок жителей, передача дел и контроль за выполнением работ.
- Реестр жильцов: Удобное хранение и управление базой данных жильцов, а также возможность отправки push-уведомлений.
- Реестр персонала: Управление информацией о сотрудниках и исполнителях работ.
- Электронное голосование и оплата счетов: Возможность для жителей подавать показания и оплачивать квитанции онлайн.
- Работа с должниками: Инструменты для повышения эффективности взыскания задолженностей от информирования до судебных исков.
- Цифровой план дома: Создание и хранение электронных планов объектов управления.

- Финансовая аналитика: Отчеты и анализ по фонду, контроль за выполнением работ и нагрузкой персонала.
- Интеграция с различными системами: Включая государственные информационные системы и другие сторонние сервисы.
- Управление задачами: Автоматизация рутинных задач и оптимизация управленческих процессов.
- Каталог дополнительных услуг: Включение новых сервисов для жителей, увеличение прибыльности работы.
- CRM-система: Автоматизированное управление взаимоотношениями с клиентами.
- Мобильные приложения: Для жителей и технических специалистов с функционалом работы в офлайн-режиме.

Кроме «Doma.ai» существует решение от «Розенталь Групп», которое предлагает следующие возможности для своих клиентов:

- Мобильное приложение и электронная диспетчерская: позволяют принимать заявки на работы, показания приборов учёта и оплаты. Проведение опросов и электронных собраний.
- Мониторинг территории: предоставляют функционал для отслеживания происходящего на территории и оценки выполнения работы.
- Удалённое управление доступом: возможность открывать шлагбаум или другие доступы с помощью системы.
- Аренда специализированного оборудования: компания предлагает в аренду разнообразное оборудование, которое необходимо для обслуживания дома, избавляя ТСЖ от необходимости постоянного его содержания в штате.
- Профессиональные услуги: предоставление услуг инженеров, теплотехников, специалистов по слаботочным системам, сварщиков, сантехников, электриков, а также юристов и бухгалтерских услуг.

– Единый договор и выбор необходимых услуг: упрощение процесса взаимодействия с контрагентами за счет возможности заключения одного договора и выбора необходимых услуг через мобильное приложение.

– Оплата по факту: фиксированная стоимость услуг и оплата через систему после выполнения работ с получением отчетных документов и детализацией расходов.

– Интеграция с 1С и обучение сотрудников: предусмотрена возможность интеграции с существующими системами учёта и проведение обучения сотрудников.

– Розенталь Групп Сервис: платформа для привлечения проверенных партнёров и контроля их работы через собственную систему распределения заявок.

## Глава 2 Логическое проектирование

Логическое проектирование - это этап создания абстрактной модели системы, который предшествует физическому проектированию. На этом этапе разрабатывается схематическое представление информационных потоков, функций системы и способов хранения данных, независимо от программных и аппаратных средств её реализации. [17]

### 2.1 Логическая модель информационной системы и её описание

Логическая модель информационной системы - это представление данных и компонентов системы, показывающая их взаимодействие между собой, без раскрытия физических деталей реализации. Для создания логической модели можно использовать UML диаграммы. [18] UML предоставляет много способов графического описания процессов и компонентов системы, но для описания работы небольшой организации, такой как правление многоквартирного дома наиболее удобной является диаграмма прецедентов. Она отражает всего 3 вида сущностей: акторы – категории действующих лиц. Один актор может подразумевать человека в определённой должности или сервис, его заменяющий. Один актор отражает поведение всех участников системы на данной должности, вне зависимости от их числа. Типичные для актора действия называются прецедентами. На диаграмме можно также уточнить тип связей между акторами и прецедентами. [19]

В контексте разработки информационной системы для управления многоквартирным домом, составление этой диаграммы помогает в составлении функциональных требований. Описанные прецеденты в дальнейшем будут реализованы в программном обеспечении, акторы будут взяты за основу для системы учётных записей.

Чтобы разработать информационную систему для помощи в управлении МКД, в начале необходимо рассмотреть всех участников процесса, не делая

акцента на возможностях именно информационных систем. Это позволит выявить дополнительные требования. На рисунке 3 представлена диаграмма вариантов использования в контексте ручной (не предусматривающего использование ИС) обработки всех возникающих проблем.



Рисунок 3 – Диаграмма прецедентов. Управление многоквартирным домом.

Единственный актер, обозначающий одного человека – это актер председателя ТСН, он отображён на рисунке 4. Председатель должен быть хорошо знаком со всеми проблемами обслуживаемого дома. В его задачи входит подготовка перечня работ по обслуживанию и ремонту. При передаче управления домом к товариществу собственников недвижимости, управляющая компания предоставляет документы необходимые для оценки технического состояния дома. Этой оценкой занимается (или руководит) председатель. После осмотра общего имущества и изучения документов, выявляются неисправности. Председатель фиксирует их, затем составляет

перечень работ по их устранению. Кроме мелкого, среднего или капитального ремонта дома, необходимо проводить ежегодное обслуживание, подготовку к отопительному периоду и другие повторяющиеся из года в год мероприятия. Для учёта начала и завершения таких работ необходимо иметь календарный план. Его заполнение и отслеживание также входит в обязанности председателя. В случае отсутствия автоматизации деятельности ТСН, календарный план ведётся вручную в бумажном журнале или в электронных таблицах. Это позволяет надёжно фиксировать все данные, но требует времени на ручную обработку и не может напомнить о приближении заявленных сроков выполнения ремонтных работ.

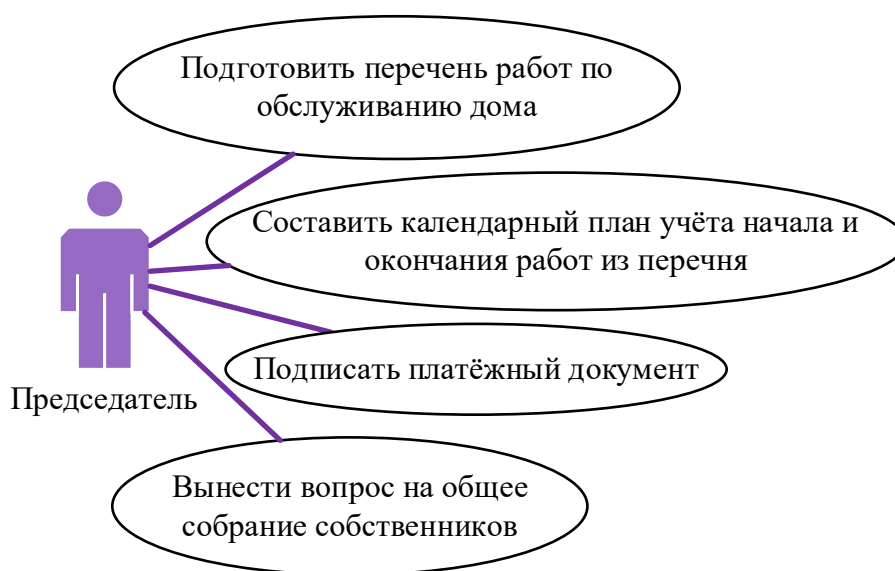


Рисунок 4 – Председатель на диаграмме прецедентов.

После председателя, значимым элементом руководства является правление, актер которого представлен на рисунке 5. В его обязанности входит контроль выполнения работ по ремонту или обслуживанию дома. Для контроля сроков исполнения правлению необходим календарный план работ, заполненный председателем. При отсутствии информационной системы правлению необходимо запоминать все мероприятия, что увеличивает вероятность забывания при большом количестве мероприятий. АИС же может отправлять уведомления о приближении сроков всех записанных



мероприятий. Перед заключением договоров с подрядчиками, необходимо согласие жильцов, так как именно их деньги будут расходоваться на оплату работ. При работе без использования АИС, правление вынуждено писать бумажные письма для жильцов и оставлять их в почтовых ящиках и вывешивать листовки с напоминанием о собрании на доске информации дома. Такой подход не гарантирует оповещение всех жильцов.

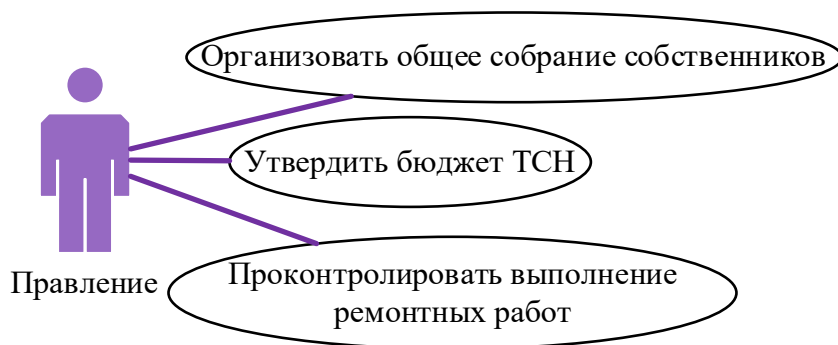


Рисунок 5 – Правление на диаграмме прецедентов

Для недопущения излишних трат бюджета, кроме аппарата управления в структуре ТСН предусмотрен надзорный орган, называемый ревизионной комиссией. На рисунке 6 представлены основные задачи, решаемые ревизионной комиссией. Для выполнения задач комиссии требуется доступ к документам и финансовым отчётам товарищества. При отсутствии АИС, все расчёты для проверки отчётов проводятся вручную, результаты проверки сохраняются в бумажном виде и хранятся в сейфе, что добавляет дополнительные бюрократические трудности по работе с ними.

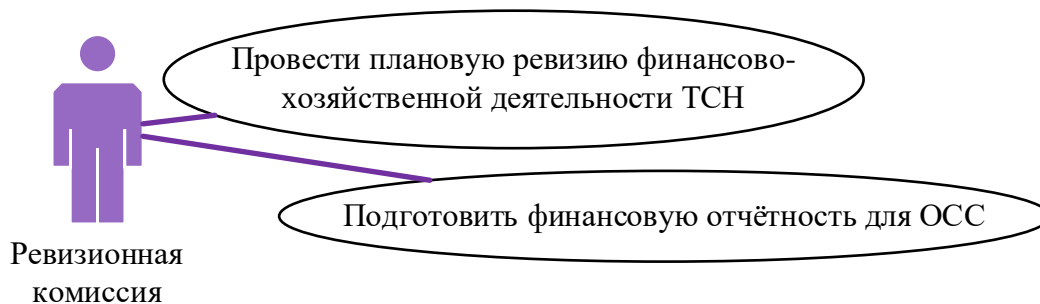


Рисунок 6 – Ревизионная комиссия на диаграмме прецедентов

Финансированием ремонта и обслуживания МКД, участием в голосовании и непосредственным пользователем услуг председателя и правления являются жильцы дома. Актор, представляющий их отображён на рисунке 7. При отсутствии информационной системы им необходимо ежедневно проверять свой почтовый ящик и внимательно просматривать доску с информацией во избежание упущения уведомления о важном событии. Собственникам квартир необходимо выделять время для посещения собраний. При неявке их голос не будет учтён, не будет возможности обсудить с правлением вопросы голосования. Возможна ситуация, когда ремонт подрядчиками проводится некачественно и недостатки не фиксируются председателем ввиду сильной загруженности. Обнаруженные изъяны возможно донести до руководства ТСН только в письменном виде или при личном обращении, что усложняет оперативное принятие мер.

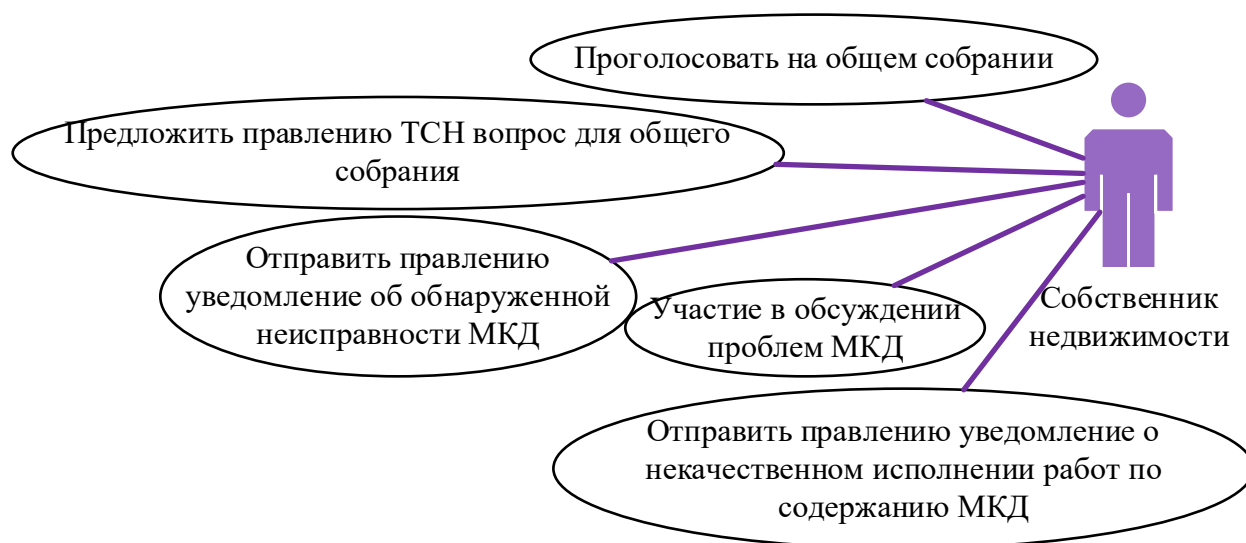


Рисунок 7 – Собственник на диаграмме прецедентов

Исполнителем большинства работ по обслуживанию дома являются коммунальные службы. В их задачи входит устранение аварий, уборка прилежащих к дому территорий и обеспечение жильцов дома водой, газом, электроэнергией. Некоторые организации могут предоставлять дополнительные услуги. Взаимодействие председателя ТСН с

ресурсоснабжающими организациями в этой работе подробно не рассматривается, так как у ООО «Квартплата24» все необходимые сервисы в этой области.

Рассмотрим возможные варианты участия прежних акторов в разрабатываемой системе. Коммунальные службы не участвуют процессах системы, следовательно, рассмотрены не будут. Председатель будет иметь свой личный кабинет в веб-версии приложения, в котором будут отображаться все необходимые ему актуальные данные. В их числе: календарный план мероприятий по обслуживанию дома, обращения жильцов, договоры о выполнении работ по ремонту и обслуживанию дома, финансовые отчёты ТСН и все документы, переданные управляющей компанией. Личный кабинет будет предоставлять некоторый функционал, представленный в виде прецедентов на рисунке 8.



Рисунок 8 – Функции личного кабинета председателя.

Председатель отвечает за заполнение календарного плана мероприятий по ремонту и обслуживанию дома. В рассмотренном ранее варианте работы ТСН, записи о мероприятиях производились на бумажном носителе, что создавало множество неудобств для дальнейшей работы с ними. При переходе на информационную систему, данные о сроках всех предстоящих ремонтов дома становятся доступными для всех жильцов в режиме реального времени. Права доступа к этим данным ограничены в соответствии с ролью в управлении МКД. В личном кабинете также будет страница, адаптированная под мобильные устройства, предназначенная для заполнения перечня неисправностей дома. Такой подход позволит председателю во время осмотра дома делать записи о дефектах, используя мобильный телефон. Остальные разделы личного кабинета рассчитаны на работу за компьютером. Просмотр и добавление данных с разных устройств производятся в общей базе данных. Обращения, написанные собственниками квартир через бот, будут сохранены и доступны председателю. Председатель сможет не только просматривать обращения, но и отвечать на них. Ответы, отправленные из веб-клиента, будут доставляться в виде личных сообщений от бота. На основе перечня выявленных неисправностей и полученных обращений жильцов, председатель составляет повестку голосования. Само голосование можно провести очно и дистанционно, затем объединить данные о голосах и получить рассчитанные системой результаты голосования. АИС также будет уведомлять всех жильцов дома о дате и времени предстоящего очного собрания. Результаты голосования отображаются в личном кабинете председателя и рассылаются всем членам ТСН.

Ревизионная комиссия, как и в случае отсутствия автоматизации занимается контролем финансовых операций товарищества на основе его отчётов. Её основные функции представлены на рисунке 9. Внедрение информационной системы позволяет перейти от ручных расчётов к заполнению электронных форм и просмотру результатов. Для учётной записи сотрудника предусматривается собственный набор прав доступа к важным

документам, что позволяет не подвергать риску сохранность важной информации, как это возможно при использовании бумажных журналов. При достаточно высоком уровне автоматизации контроля отчётности в системе, необходимость в сотрудниках ревизионной комиссии может полностью исчезнуть. В таком случае уведомления о несоответствии будут отображаться в личном кабинете председателя и рассылаться всем жильцам. Полная автоматизация проверки отчётности позволит сократить бюджет ТСН и увеличить прозрачность управления МКД.

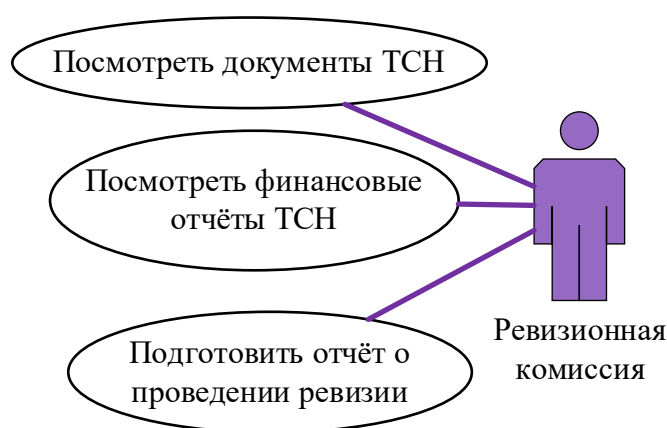


Рисунок 9 – Функции ревизионной комиссии.

Кроме ревизионной комиссии, автоматизации частично будет подвержено и правление товарищества. Его обязанности представлены на рисунке 10. Одна из обязанностей – уведомление жильцов о предстоящем собрании. При отсутствии АИС, эта задача требует значительного количества времени на создание брошюр с повесткой собрания и их вручение. После внедрения системы, уведомления будут приходить жильцам сразу после заполнения перечня вопросов собрания собственников и назначения времени его проведения. После подсчётов результатов голосования, при ручном подходе, правлению приходилось выдавать их на руки всем жильцам, с целью увеличения числа ознакомившихся. При этом возможна ситуация, когда часть жильцов не проживает дома в период проведения голосования, но по возвращении, изъявляет желание ознакомиться с результатами, хранящимися

в сейфе управления ТСН. Информационная система рассылает результаты автоматически после их подсчётов. Правление также взаимодействует с информационной доской дома. В случае использования автоматизированной информационной системы, вместо доски размещённой в подъезде дома, используется веб-версия информационного стенда. Такой способ ознакомления более удобен для современных жильцов, так как позволяет просматривать записи о мероприятиях из любого места в любое время при наличии соединения с интернетом. Такой источник информации проще поддерживать в актуальном состоянии, так как редактировать его можно даже не посещая МКД. В обязанности правления также входит подготовка отчётов об использовании бюджета. При ручном подходе к управлению МКД, такая отчётность может быть только в бумажном виде, что создаёт трудности у собственников, желающих с ней ознакомиться.



Рисунок 10 – Обязанности правления ТСН

Последней категорией пользователей проектируемой системы является собственник недвижимости. Из рисунка 11 можно выделить основные возможности системы, которые необходимы данной категории пользователей. Так как собственники оплачивают ремонт и обслуживание, то им важно иметь представление о расходе вложенных ими средств. В отличие от бумажной

отчётности, получить доступ к которой можно только при личной встрече с любым из членов правления или председателем, электронная отчётная документация доступна для ознакомления в любое время. Права на доступ в целях безопасности также ограничиваются системой. Собственник может принимать участие в голосовании, о котором будет своевременно уведомлён системой. При проведении правлением голосований дистанционного формата, жильцы получают через бота последовательность опросов, представляющих вопросы из повестки голосования. После завершения голосования, всем собственникам системой рассылается сообщение с результатами. Используя функционал бота, пользователи также смогут отправлять обращения, идеи, жалобы на качество ремонта председателю, просмотреть календарный план работ по обслуживанию дома. Удобство бота для жильцов обусловлено использованием одного приложения для общения и взаимодействия с системой. Программа для обмена сообщениями привычна для неподготовленных пользователей так как часто используется ими. Функционал бота возможно реализовать в виде веб или мобильного приложения, но это вызовет дополнительные неудобства у потенциальных пользователей. Мобильное приложение необходимо устанавливать смартфон, что может быть невозможно из-за несовместимости с операционной системой или нехватке памяти на устройстве. Веб-приложение потребовало бы открытия в браузере, что могло бы вызвать некорректное отображение содержимого и непредвиденные ошибки из-за плохой адаптированности приложения под конкретный браузер пользователя.



Рисунок 11 – Функционал системы для собственника

Выше описаны все основные функциональные требования к проектируемой системе, полный их перечень представлен в виде диаграммы прецедентов на рисунке 12.



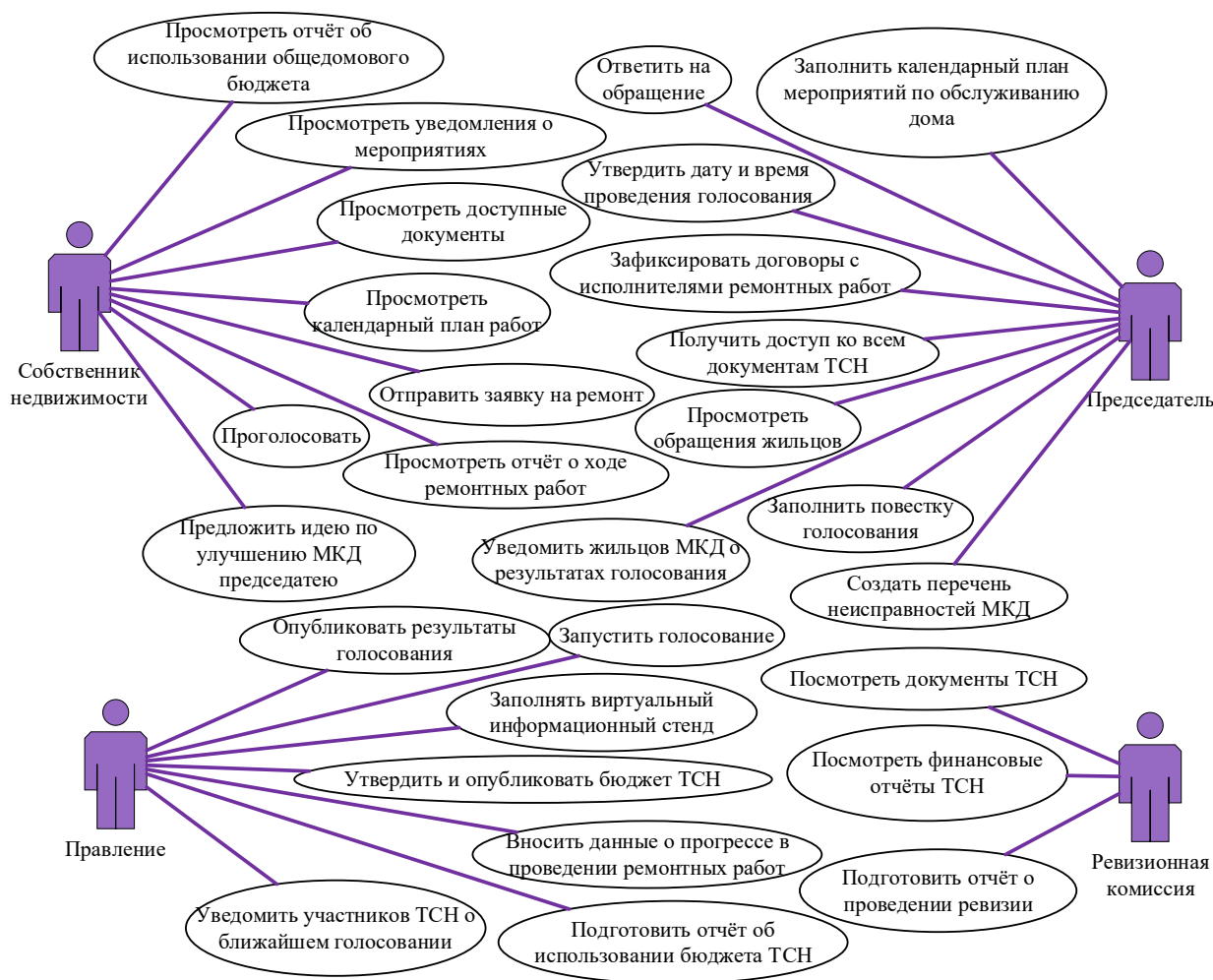


Рисунок 12 – Диаграмма прецедентов. Использование ИС в процессе организации деятельности ТСН

Действия разных участников системы происходят в определённой последовательности, которая подробно отражена на рисунке 13. Считается что автоматизация ведения и проверки отчётности в системе достаточная для отказа ТСН от ревизионной комиссии. Её работа полностью выполняется автоматизированной информационной системой. На диаграмме представлены пулы потоков, соответствующие акторам из прошлых диаграмм: председатель, правление, собственник, и отдельно рассмотрена в качестве исполнителя действий. [20] Таким образом показано, что она является не просто связующим звеном или удобным инструментом, а полноценным участником общих процессов. Рассмотрены последовательности процессов

всех пользователей системы от момента запуска системы, до исчерпания требуемого функционала для каждого из пользователей.

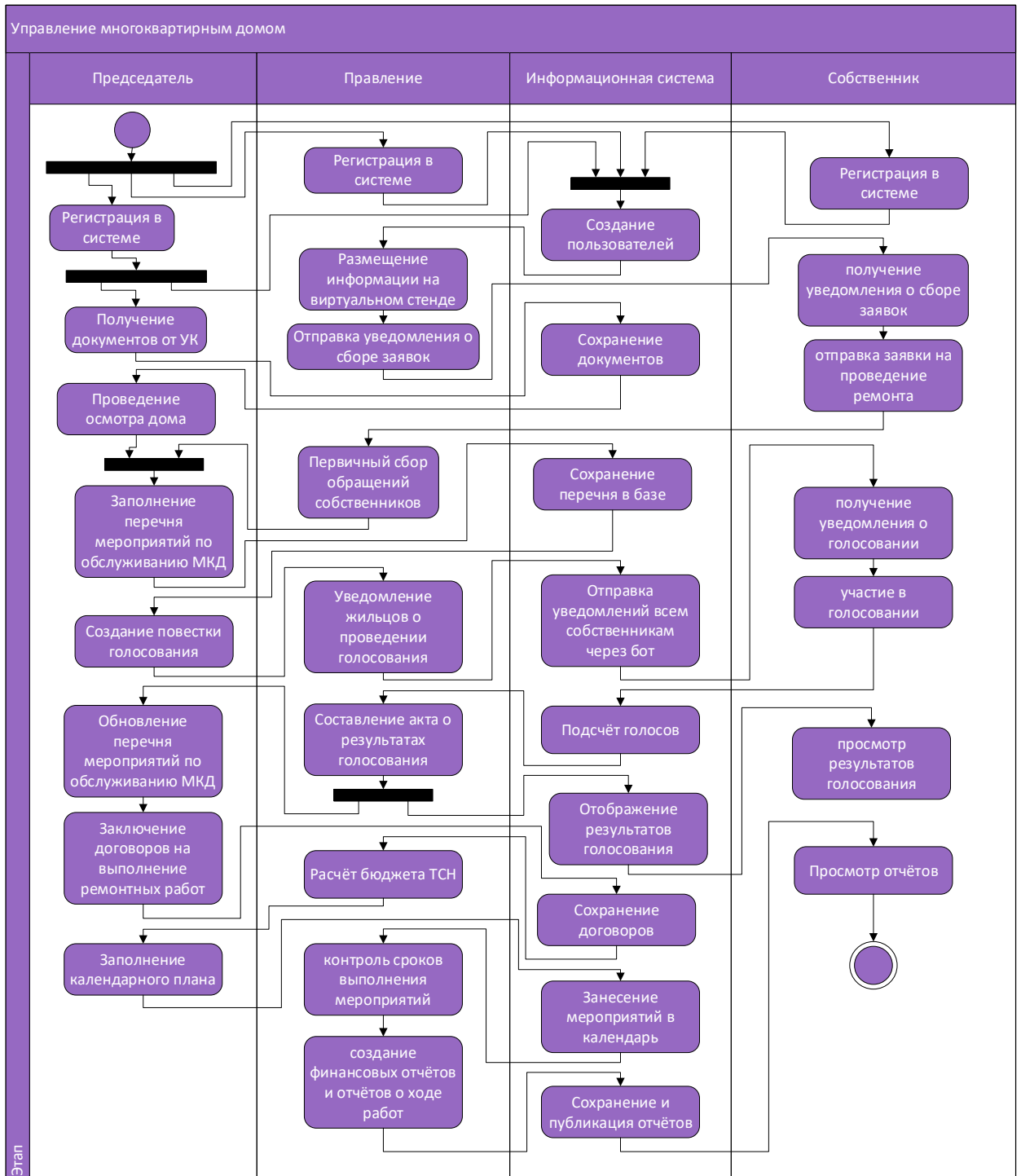


Рисунок 13- Управление МКД на диаграмме деятельности

Работа в системе начинается с регистрации каждого пользователя, что можно считать параллельными событиями.

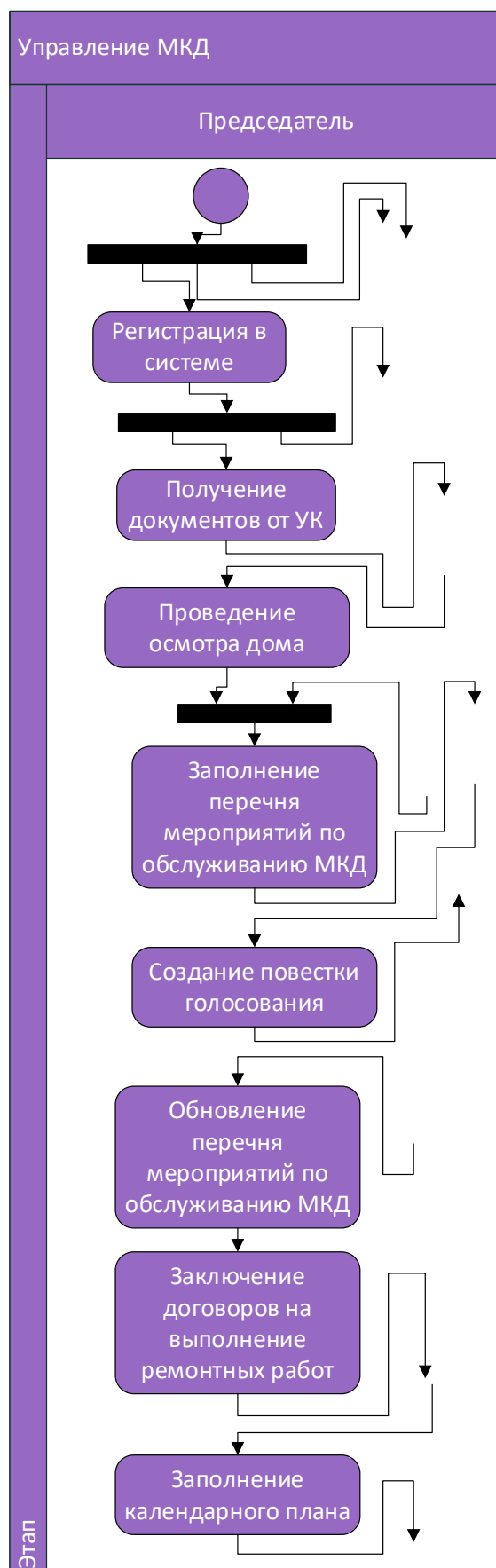


Рисунок 14 – Председатель на диаграмме деятельности

Для работы с отчётами, проведения осмотров МКД на наличие неисправностей, и дальнейшего составления дефектных ведомостей председателю необходимо получить от управляющей компании все документы и инженерные планы дома. Последовательность его действий представлена на рисунке 15. Все полученные документы председатель сохраняет в системе и, сверяясь с ними проводит осмотр дома.

Поддерживать цифровую информационную доску в актуальном состоянии входит в обязанности правления. Перед проведением собрания полезно изучить мнение жильцов о состоянии дома и приоритетах его ремонта. Для этого необходимо объявить о сборе обращений в электронном виде, затем обработать результаты и предоставить отчёт председателю. После принятия решения о приоритетах ремонта дома составляются вопросы для голосования. Так как голосование с использованием информационной системы проводится через бот, необходимо, чтобы всем его участникам пришло уведомление с указанием сроков проведения. Система отправляет уведомления автоматически, но утвердить сроки требуется от правления. Правление также осуществляет расчёт бюджета с использованием ИС.

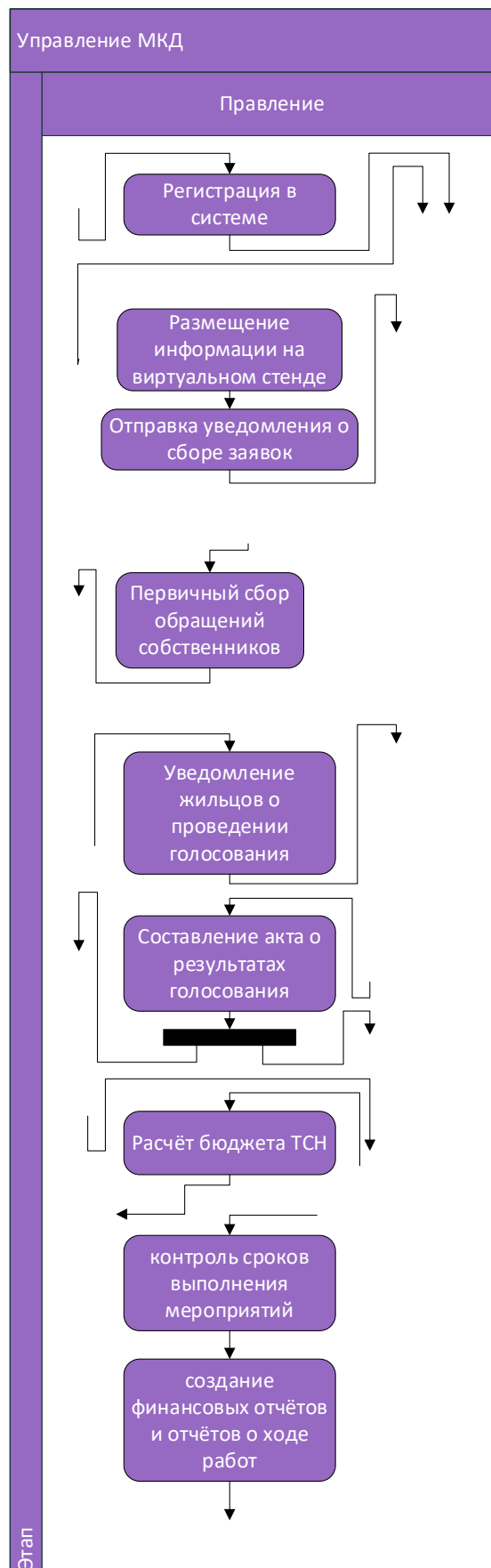


Рисунок 15 – Правление на диаграмме деятельности

При проектировании системы важно рассмотреть все классы, которые можно условно поделить на группы: классы пользователей, классы документов и классы событий.

Для хранения документов нужно предусмотреть родительский класс, от которого можно наследовать классы документов различных типов. Родительский класс должен иметь скрытые поля с его идентификатором, названием документа, типом документа, пути его расположения в файловом хранилище системы, и датой появления этого документа в системе. [21] Методы должны предоставлять функционал для получения информации о документе, загрузки документа в систему, скачивания его на компьютер.

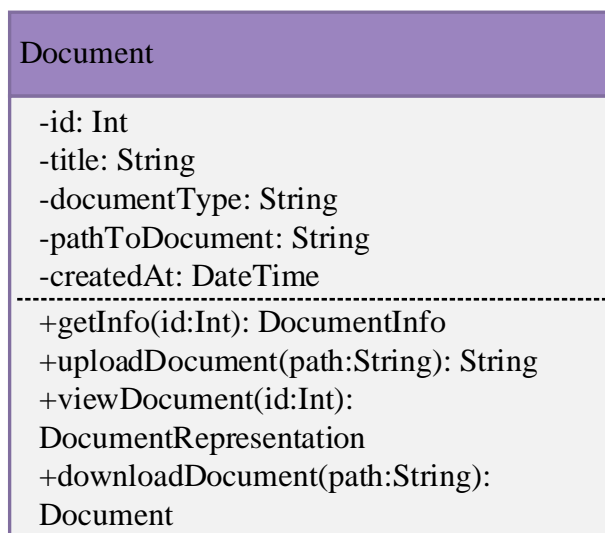


Рисунок 16 - Класс документ

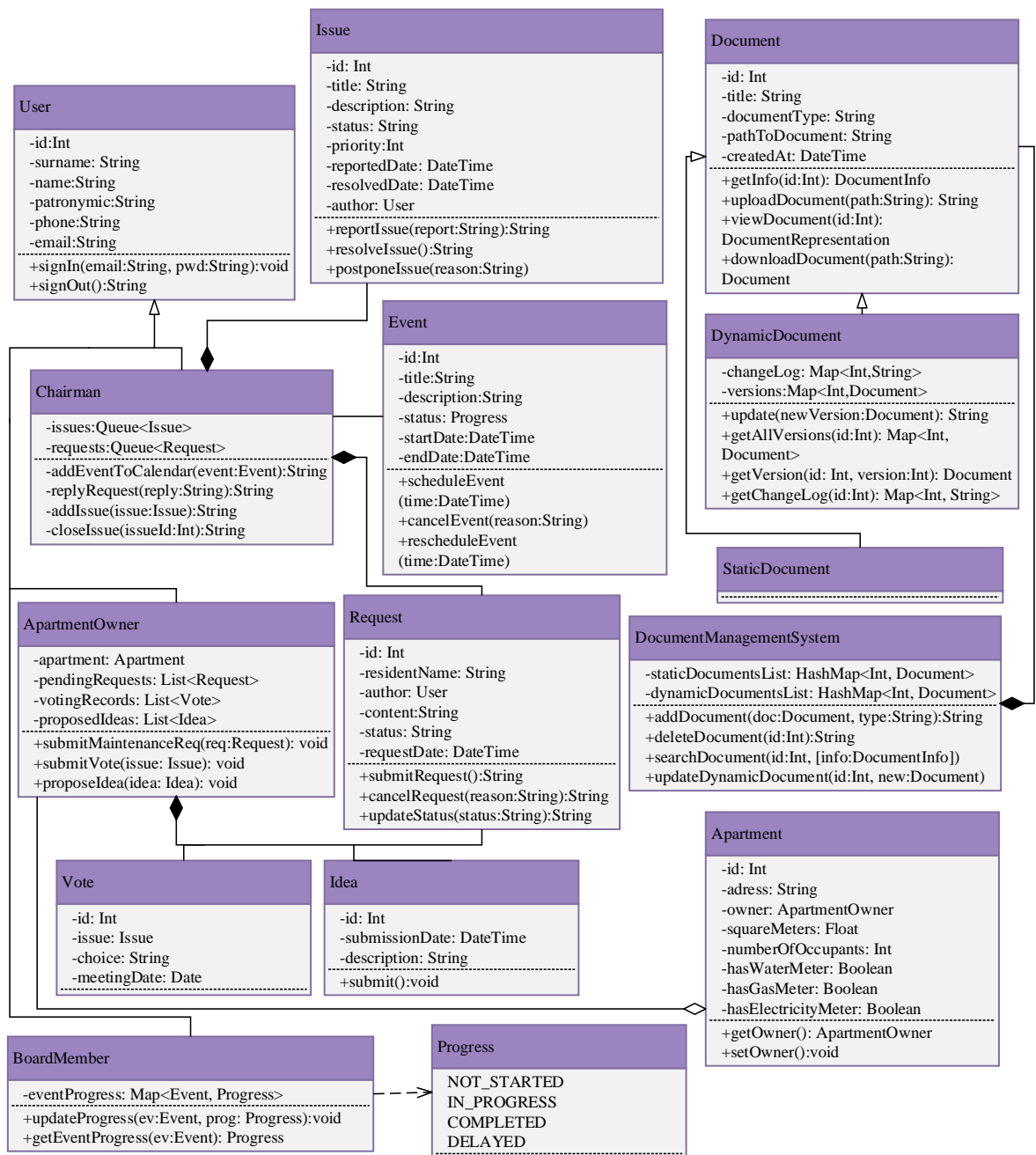


Рисунок 17 - Диаграмма классов

В системе управления документами и запросами жилого комплекса рассмотрим подробное описание классов и их взаимоотношений.

Класс User отражает сущность пользователя системы, предоставляя идентификатор (ID), имя, уникальное имя пользователя, адрес электронной почты, пароль и роль пользователя, которая может быть, как «житель», так и «арендодатель». Этот класс служит основой для взаимодействия с функционалом системы в соответствии с ролью пользователя.

Класс Issue описывает проблему, поднимаемую жителем для арендодателя с привязкой к идентификатору, названию, описанию, приоритетности, статусу решения, а также дате и времени подачи заявления, и дате, и времени решения проблемы. Это позволяет эффективно управлять проблемными вопросами в рамках комплекса.

Класс Event предназначен для отображения событий, происходящих в жилом комплексе. Он включает в себя такие поля, как идентификатор, название события, его описание и временные рамки проведения. Это важный элемент для организации общественной активности и информирования жителей.

ApartmentOwner представляет информацию о владельце квартиры, содержа идентификатор, имя, контактный номер телефона и адрес электронной почты. Это позволяет упростить связь между управляющими комплексом и владельцами квартир.

Request определяет запросы, отправляемые жителями к арендодателям или управляющим, включая идентификатор, название, описание, статус запроса, а также дату и время создания и закрытия запроса. Этот механизм способствует более организованному процессу обращения и решения текущих вопросов.

Idea класс создан для инициирования идей от жителей, которые они хотели бы предложить для улучшения жизни в комплексе. Содержит поля для идентификации, описания идеи, её статуса и времени подачи.

BoardMember описывает членов правления жилого комплекса, их идентификаторы, имена и роли в управлении комплексом. Это важно для разграничения ответственности и полномочий внутри административной структуры.

Progress используется для отслеживания прогресса по запросам или проблемам, предоставляя текущий статус и время обновления статуса. Это обеспечивает прозрачность процесса обработки обращений.



Document класс отвечает за документы внутри системы, позволяя хранить идентификатор, название, описание, путь к файлу и дату создания документа. Это упрощает управление документооборотом в комплексе.

DocumentManagementSystem представляет систему управления документами и запросами, включая комплекс методов для работы с документами: добавление, обновление, удаление, поиск и получение представлений документов. Это ядро системы, которое обеспечивает её функциональность.

Apartment класс, описывающий квартиры в комплексе через идентификатор, адрес, площадь, количество жителей и данные по коммунальным показаниям (водомер, газомер и электросчетчик). Это позволяет администрировать жилые площади и учитывать особенности каждой квартиры.

## **2.2 Разработка логической модели данных**

Для функционирования системы требуется работа с данными. В разрабатываемой системе удобно использовать реляционные СУБД, в частности PostgreSQL. В базе данных должна быть разработана структура, позволяющая хранить состояния классов. Отношения между таблицами регулируются внешними ключами, что обеспечивает дополнительную безопасность при работе системы, так как не допускает выполнения операций, которые затрагивают связанные таблицы. [9]

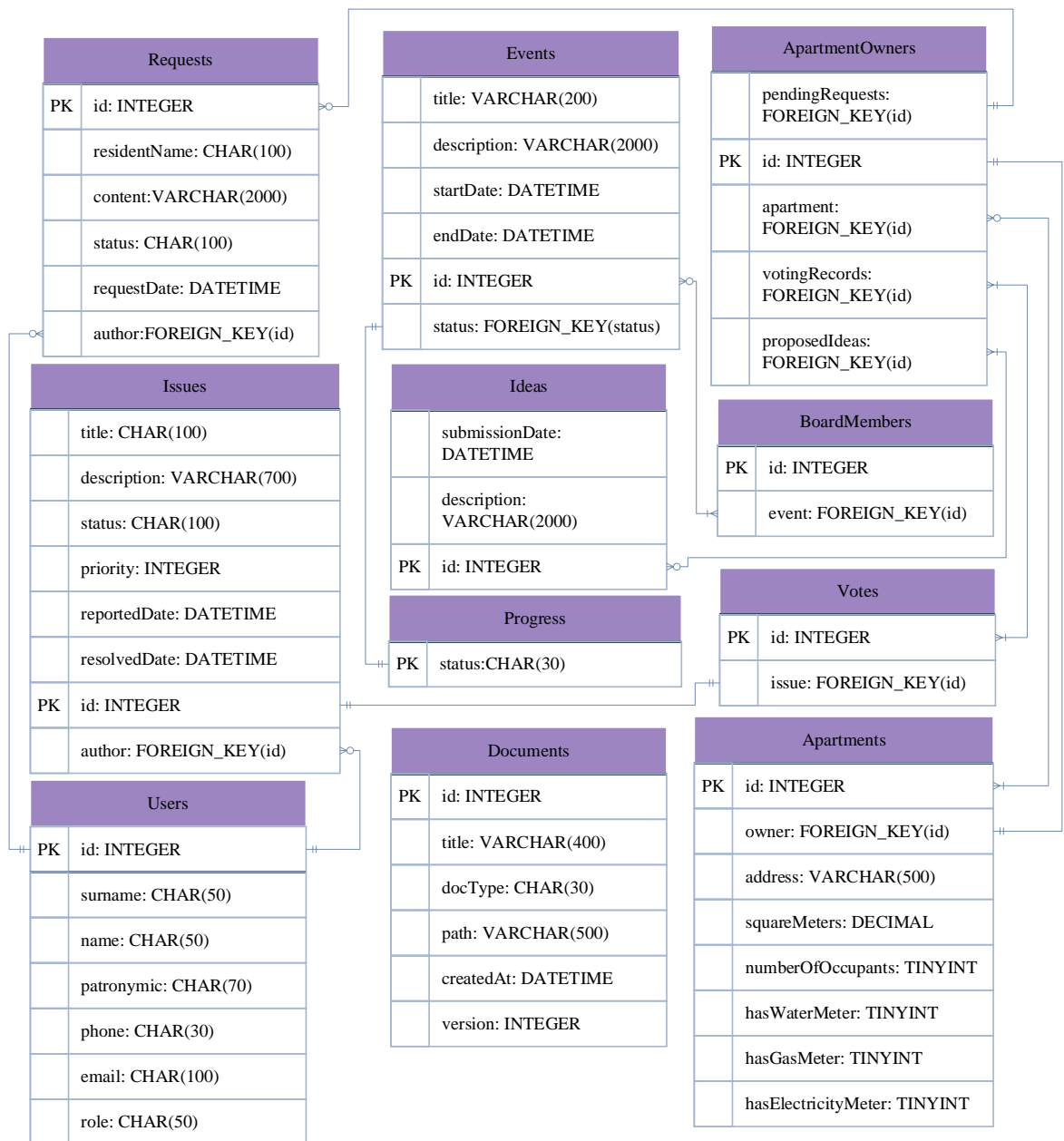


Рисунок 18 – Логическая модель данных

## Глава 3 Тестирование информационной системы

### 3.1 Модульное тестирование

Разработанное приложение состоит из нескольких основных модулей, поэтому для проверки корректности работы системы в целом, нужно проверить как работоспособность каждой её части, так и взаимодействие между ними. Для проверки функциональности хорошо подходят unit-тесты, так как позволяют проверять работу методов изолированно. Веб-приложение взаимодействует с СУБД Postgresql. Полученные из базы данные обрабатываются и выводятся на шаблоны веб страниц. Полученные из форм данные сохраняются в базе. Если на этапе взаимодействия с СУБД появятся ошибки, они будут распространяться на все последующие архитектурные слои. Непосредственно к базе данных обращается слой Repository. Необходимо проверить корректность получения данных и их добавления, при этом не изменяя состояние действующей базы данных. Для решения этой проблемы подходит использование тестовой СУБД, которая создаётся отдельно и существует в оперативной памяти, без сохранения на жёсткий диск только во время проведения тестирования. Наиболее популярным решением является H2 Database Engine. H2 это популярная встраиваемая база данных, разработанная Томасом Мюллером в 2005 году, обладающая широкой поддержкой SQL стандартов и предлагает поддержку JDBC API.

```
✓ Tests passed: 1 of 1 test – 823 ms
Hibernate:
    select
        e1_0.id,
        e1_0.description,
        e1_0.end_date,
        e1_0.is_house_service,
        e1_0.start_date,
        e1_0.status,
        e1_0.title
    from
        events e1_0
    where
        e1_0.id=?
Hibernate:
    insert
    into
        events
        (description, end_date, is_house_service, start_date, status, title)
    values
        (?, ?, ?, ?, ?, ?)
```

Рисунок 19 – проверка выполнения SQL запросов

Перед тестированием репозитория создаётся подключение к БД H2 и выполняются запросы на вставку и извлечение данных, логи которых отображены на рисунке 19. В разработанном приложении используется ORM система Hibernate, которая отвечает за создание таблиц в базе данных и проведении всех операций по добавлению или получению данных. На рисунке 19 видно, что запросы были выполнены без ошибок, таким образом проверена корректность работы Hibernate.

```
j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory
t.repositories.EventRepositoryTest      : Started EventRepositoryTest in 4.27 seconds
tsj_help.services.EventServiceTest     : тест запущен
tsj_help.services.EventServiceTest     : Проверка сохранения события в базе

tsj_help.services.EventServiceTest     : событие сохранено
tsj_help.services.EventServiceTest     : событие получено
j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit
```

Рисунок 20 – Тестирование Repository

После запуска СУБД H2 и подключения к ней Hibernate можно запускать unit тесты для проверки взаимодействия с базой данных на уровне репозитория. На рисунке 20 представлены логи проведённых тестов, по которым видно, что сохранение события в базе и последующее его извлечение прошли успешно. Тестирование репозитория завершено успешно.

Следующий архитектурный слой называется сервисным. В нём обычно реализуется обработка полученных от репозитория данных и передача их в контроллеры. В разработанной системе сервисный слой выступает в качестве промежуточного, но не содержащего сложной обработки данных. Результаты unit-тестирования сервиса событий представлены на рисунке 21.

```
INFO tsj_help.services.EventServiceTest -- Тест запущен
INFO tsj_help.services.EventServiceTest -- Проверка выдачи всех событий из базы
INFO tsj_help.services.EventServiceTest -- Возвращён правильный список событий
INFO tsj_help.services.EventServiceTest -- Тест запущен
INFO tsj_help.services.EventServiceTest -- Проверка добавления нового события
INFO tsj_help.services.EventServiceTest -- событие добавлено
INFO tsj_help.services.EventServiceTest -- Тест запущен
INFO tsj_help.services.EventServiceTest -- Проверка выдачи событий сгруппированных по номерам месяцев
INFO tsj_help.services.EventServiceTest -- Коллекция номеров месяцев и их событий возвращена корректно
```

Рисунок 21 – Тестирование методов класса EventService

Как видно из логов, представленных на рисунке 21, были последовательно запущены unit-тесты для каждого из методов класса. Ни один из них не выдал ошибку, что свидетельствует об исправной работе всех методов. Созданные unit-тесты проверяют соответствие ожидаемых результатов возвращённым. В случае нарушения работы любого из методов будет выведено сообщение об ошибке и ожидаемом правильном поведении метода, например, как это показано на рисунке 4.

```

@Test
void getAllEvents() {
    logger.info("Проверка выдачи всех событий из базы");
    List<Event> expectedEvents = fakeEvents;
    Mockito.when(eventService.getAllEvents()).thenReturn(expectedEvents);
    List<Event> actualEvents = eventService.getAllEvents();
    assertEquals(expectedEvents, actualEvents, message: "getAllEvents должен возвращать корректный список Событий");
    logger.info("Возвращён правильный список событий");
}

```

Рисунок 22 – Обработка ошибок в тестах

Возможна ситуация, когда пользователь попытается создать событие дата начала и окончания которого введены в недопустимой последовательности. Если на веб-странице эта ошибка не была обработана, то следующим этапом, на котором можно остановить создание ошибочного события является сервисный слой. На рисунке 5 представлен код и результат его тестирования при попытке создать событие с ошибочными датами.

```

public void createEvent(Event event) throws Exception {
    LocalDate startDate = event.getStartDate();
    LocalDate endDate = event.getEndDate();
    if (startDate.isBefore(endDate) | startDate.isEqual(endDate)) {
        eventRepository.save(event);
    } else {
        throw new Exception("The end date of the event cannot be before the start date.");
    }
}

```

```

INFO tsj_help.services.EventServiceTest -- Проверка на выброс исключения
при добавлении нового события с неправильной датой
INFO tsj_help.services.EventServiceTest -- Исключение успешно выброшено

```

Рисунок 23 – Выявление ошибочной даты

Сервис для обработки обращений также необходимо покрыть unit-тестами. Основные тестовые сценарии для методов класса связаны с получением обращений из репозитория и добавлением новых. При попытке получения обращения по его идентификатору может возникнуть ошибка,

связанная с отсутствием запрашиваемой записи в базе. В таком случае программа должна корректно обработать ошибку отсутствия элемента. При создании обращения в базе репозиторий должен вернуть объект записанного в базу элемента, который затем попадёт в соответствующий сервис. При обращении к репозиторию для получения всех обращений из базы, результатом должен быть список обращений. При неправильном вызове метода репозитория может быть возвращён другой тип или выброшено исключение. Все описанные выше ситуации проверены тестами, результаты выполнения которых представлены в виде логов на рисунке 5.

```
tsj_help.services.IssueServiceTest -- Тест запущен
tsj_help.services.IssueServiceTest -- Проверка получения списка обращений из базы
tsj_help.services.IssueServiceTest -- Список обращений выведен корректно
tsj_help.services.IssueServiceTest -- Тест запущен
tsj_help.services.IssueServiceTest -- Проверка создания обращения
tsj_help.services.IssueServiceTest -- Обращение создано успешно
tsj_help.services.IssueServiceTest -- Тест запущен
tsj_help.services.IssueServiceTest -- Проверка получения обращения по id из базы
tsj_help.services.IssueServiceTest -- Возвращено корректное значение
tsj_help.services.IssueServiceTest -- Тест запущен
tsj_help.services.IssueServiceTest -- Проверка выброса исключения при отсутствии обращения в базе
tsj_help.services.IssueServiceTest -- Исключение выбрасывается корректно
```

Рисунок 24 – Тестирование сервиса обращений

По правилам пользования системой, жильцы должны делать подробное описание проблемы. Система позволяет контролировать исполнение этого требования, проверяя длину строки в объекте перед отправкой в базу данных. На рисунке 7 представлен фрагмент программного кода с реализацией проверки заполнения поля и лог соответствующего ему unit-теста. Базовые условия проверки требуют, чтобы поле описания не было пустым, и его длина превышала 10 символов. При попытке сохранения незаполненного описания, системой выбрасывается исключение.

```
public Issue saveIssue(Issue issue) throws Exception {
    Issue newIssue;
    if (!(issue.getDescription().isEmpty() | issue.getDescription().length() < 10)) {
        newIssue = issueRepository.save(issue);
    } else {
        throw new Exception("Слишком короткое описание");
    }
    return newIssue;
}
```

```
tsj_help.services.IssueServiceTest -- Проверка создания обращения с незаполненным описанием
tsj_help.services.IssueServiceTest -- Выброшено верное исключение
```

Рисунок 25 – Выявление незаполненного описания

Реализованный подход защищает от добавления некорректных данных в базу, но пользователь, который пытается эти данные добавить не узнает об ошибке при добавлении и о причинах этой ошибки. Чтобы сделать систему более информативной, для веб страницы был реализован механизм валидации данных формы. При попытке создания события с пустыми полями или при некорректном вводе дат начала и завершения события, председатель получит уведомление о некорректности данных. Пример таких сообщений представлен на рисунке 8.



The image shows a web form with several fields and a modal notification. The form fields are:

- Название:** An empty text input field.
- Дата начала:** A date picker showing "дд.мм.гггг --:--".
- Описание:** A text area with a validation error: "Вы пропустили это поле." (You missed this field).
- Дата окончания:** A date picker showing "дд.мм.гггг --:--".
- Статус:** A text area with a validation error: "Вы пропустили это поле." (You missed this field).
- Дата окончания:** A date picker showing "дд.мм.гггг --:--".
- Относится к обслуживанию дома?:** A radio button group with options "Да" (selected) and "Нет". A validation error is present: "Выберите один из вариантов." (Select one of the options).

A modal notification box is displayed in the foreground with the following text:

Уведомление от сайта localhost  
 Дата начала должна быть раньше даты окончания события.  
 [Закреть]

Рисунок 26 – уведомления о некорректности ввода

Функционал системы разделён на сервисы, которые взаимодействуют по протоколу http отправляя GET и POST запросы для обмена данными. При таком подходе появляется вероятность возникновения большого числа проблем на всех этапах, от отправки запроса до извлечения его данных. Ошибки могут возникать при формировании тела запроса, при его отправке, передаче по сети и при получении на стороне другого сервиса. Даже принятый запрос может содержать некорректные данные, которые не могут быть десериализованы. Для проверки исходящих от Spring-backend запросов требуется запустить внешний сервер, который бы принимал запросы на добавление данных и отправлял в ответ статус запроса. Для простого развёртывания такого тестового сервера подойдёт фреймворк FastAPI для языка python. Код позволяющий протестировать отправку POST запроса на создание данных, представлен на рисунке 9.

```

from fastapi import FastAPI, Request
from pydantic import BaseModel

app = FastAPI()

# Определение модели для входных данных нашего API
class FormData(BaseModel):
    |   issueId: str
    |   authorId: str
    |   responseText: str

# Эндпоинт для имитации ответа бота
@app.post("/issueAnswer")
async def issue_answer(form_data: FormData):
    |   data = {
    |       |   "issueId": form_data.issueId,
    |       |   "authorId": form_data.authorId,
    |       |   "responseText": form_data.responseText
    |   }
    |   print(data)
    |   return {"200: ОК"}

```

Рисунок 27 – Сервер для тестирования post-запроса

Ответ на обращение, написанный в личном кабинете предпринимателя должен быть отправлен написавшему это обращение пользователю python-бота. Имитировать приём данных в целях тестирования будет показанный выше сервер. Логи тестового сервера при отправке ответов на обращения представлены на рисунке 10. Они показывают, что отправляемые данные десериализуются без ошибок. Так как преобразование строки в объект является последним этапом в процессе передачи, то оно свидетельствует об успешном завершении всех предшествующих этапов. Логи также отображают данные запроса для проверки их корректности: идентификатор автора обращения, номер обращения, текст ответа от председателя и статус запроса.

```

{'issueId': '3', 'authorId': '3', 'responseText': 'Ремонт лавочек будет вынесен на голосование'}
INFO: 127.0.0.1:58239 - "POST /issueAnswer HTTP/1.1" 200 OK
{'issueId': '4', 'authorId': '1', 'responseText': 'Заказали спил дерева, во вторник придет машина'}
INFO: 127.0.0.1:58248 - "POST /issueAnswer HTTP/1.1" 200 OK
{'issueId': '6', 'authorId': '4', 'responseText': 'Ремонт ливнёвки вынесен на голосование'}
INFO: 127.0.0.1:58263 - "POST /issueAnswer HTTP/1.1" 200 OK

```

Рисунок 28 – Логи тестового сервера.

Обращения на Spring-backend поступают от бота аналогичным способом, то есть с помощью HTTP POST-запроса. Проверить работоспособность их обработчика можно с помощью программы Postman. Интерфейс программы с заполненными полями для отправки запроса показан на рисунке 11.

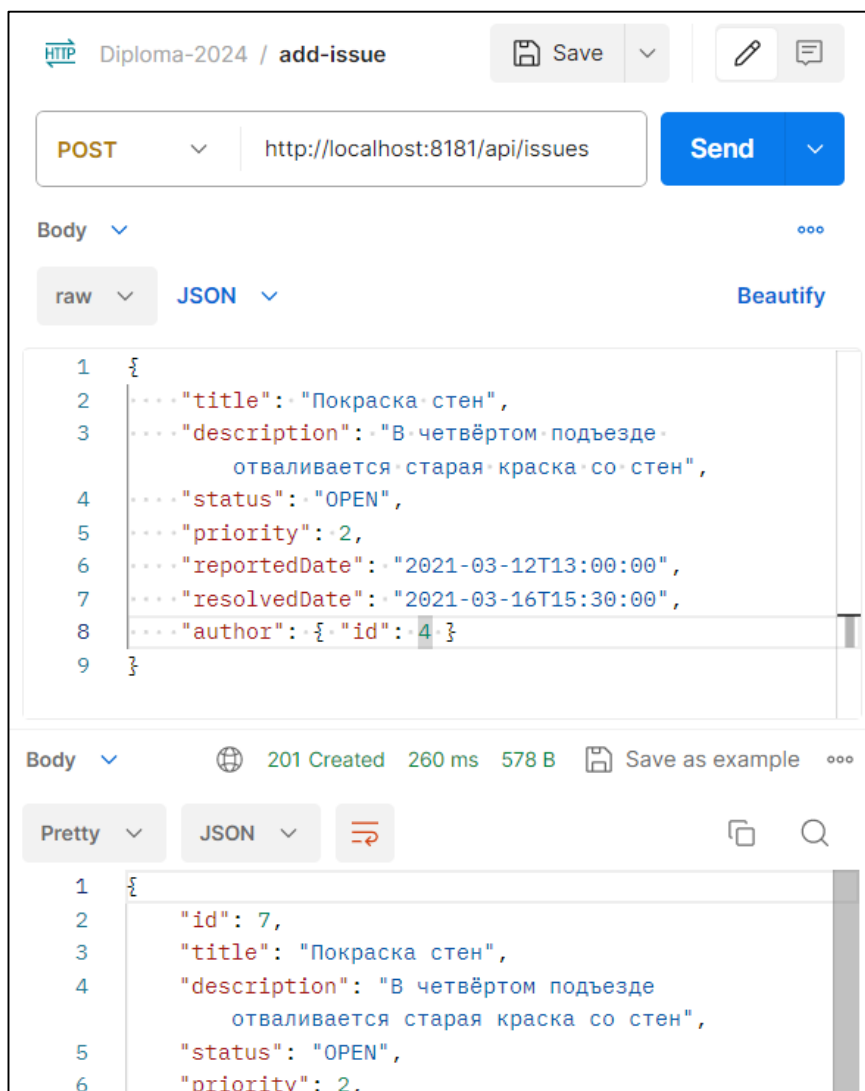


Рисунок 29 – Запрос в программе Postman

Отправленный запрос вернул статус 201, что свидетельствует об успешном его принятии. Кроме того, был получен ответ, в котором кроме отправленных данных присутствует идентификатор обращения, присвоенный при сохранении обращения в базе данных. Проверить был ли элемент добавлен можно простым сравнением содержимого соответствующей

таблицы до и после выполнения запроса. Это сравнение приведено на рисунке 12.

WHERE		ORDER BY	WHERE		ORDER BY
id	description		id	description	
1	1 в 34 квартире в ванной сломался кран		1	1 в 34 квартире в ванной сломался кран	
2	2 на 9 этаже в 3 подъезде капает с потолка		2	2 на 9 этаже в 3 подъезде капает с потолка	
3	3 лавочки около второго подъезда все слома...		3	3 лавочки около второго подъезда все слома...	
4	4 Дерево около первого подъезда сухое дере...		4	4 Дерево около первого подъезда сухое дере...	
5	6 Около второго подъезда повреждён водосли...		5	6 Около второго подъезда повреждён водосли...	
6	5 сломан доводчик на подъездной двери		6	5 сломан доводчик на подъездной двери	
			7	7 В четвёртом подъезде отваливается старая...	

Рисунок 30 – Сравнение состояний таблицы

При сравнении видно, что новый элемент действительно был добавлен. Это дополнительно подтверждает корректную работу обработчика запросов и правильность структуры отправленного запроса.

Последним компонентом системы, который требуется протестировать является telegram-bot. Для приёма запросов от него был развёрнут тестовый сервер на фреймворке fastapi. Бот должен отправлять на сервер обращения, которые будут доступны председателю. Проверим работу бота. На рисунке 13 показан пример заполнения обращения с использованием telegram-бота.

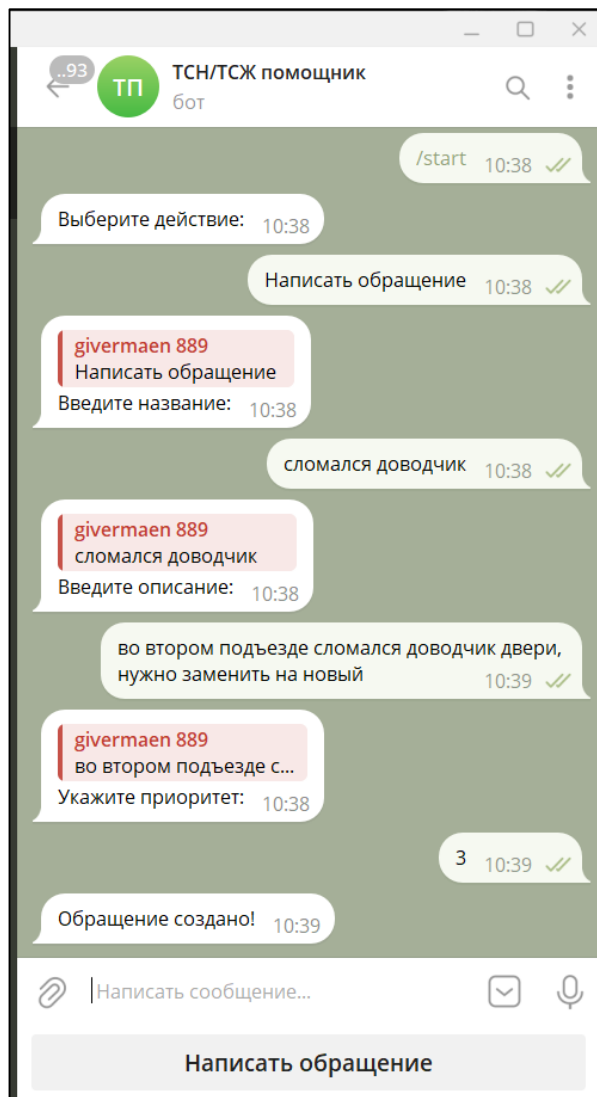


Рисунок 31 – Заполнение обращения

Для отправки обращения через бот, необходимо ввести краткое название проблемы и более подробное её описание, после чего указать приоритет. После ввода, эти данные автоматически отправятся на сервер. Бот предложит повторить процесс написания обращения. Для проверки отправки данных можно воспользоваться подготовленным небольшим тестовым сервером. Ответ тестового сервера приведён на рисунке 14.

```
Получен запрос:
{'title': 'сломался доводчик', 'description': 'во втором подъезде
сломался доводчик двери, нужно заменить на новый', 'status': 'Ожидает
рассмотрения', 'priority': 3, 'reportedDate': '2024-05-23T10:51:48',
'resolvedDate': None, 'author': {'id':3}}
INFO:      ::1:51052 - "POST /api/issues HTTP/1.1" 200 OK
|
```

Рисунок 32 – Проверка отправки данных ботом

По сообщениям тестового сервера видно, что запрос от бота действительно был отправлен, все введённые данные переданы без изменений. Кроме данных, введённых пользователем был также передан его идентификатор, по которому будет доставлен ответ председателя.

Все компоненты системы прошли тестирование по отдельности, можно переходить к функциональному тестированию системы в целом.

### 3.2 Функциональное тестирование

На данном этапе все компоненты системы должны функционировать вместе. Кратко повторим основные функциональные требования к системе управления многоквартирным домом. Telegram-бот должен отправлять запросы к серверу для сохранения данных и принимать их от сервера, при написании председателем ответов на обращения. Жильцы должны получать от бота инструкции по заполнению обращения и сообщение об успешной его отправке. В чат, в котором было отправлено обращение должно прийти сообщение от председателя в случае его написания. Пользователей бота может быть множество, и ответы на обращения должны приходить только их отправителям. Принятые сервером обращения должны сохраняться в базу данных и отображаться на веб-странице личного кабинета председателя. Личный кабинет председателя должен отображать все обращения, события и

форму для создания новых событий. При выборе любого из обращений, должна открываться отдельная веб страница с подробным описанием проблемы и полем для ответа. Собственникам квартир дома должна быть доступна страница для просмотра событий, относящихся к обслуживанию дома. События на этой странице должны быть сгруппированы по месяцам. Проверим соответствие разработанной системы поставленным требованиям.

Запустим бот и заполним обращение. После ввода всех данных, сервер отправляет сообщение об успешно отправленном обращении. Заполнение и отправка обращения представлены на рисунке 15.

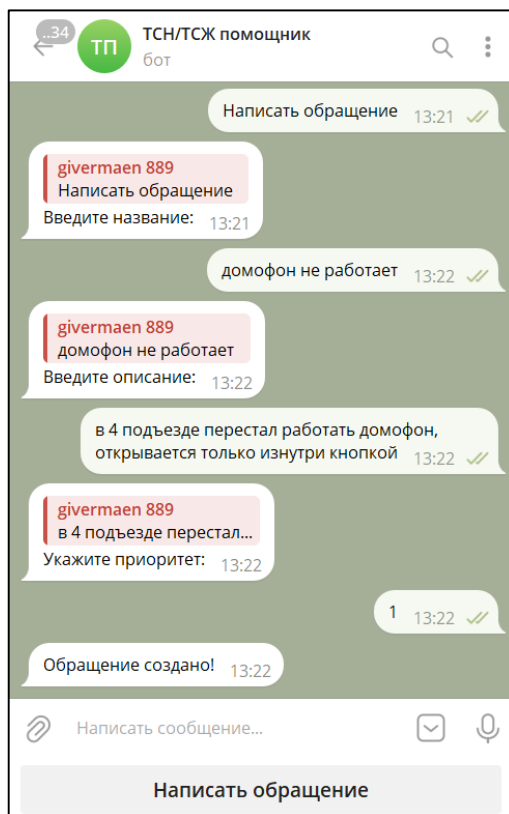


Рисунок 33 – Заполнение обращения

Теперь проверим наличие обращения в личном кабинете председателя. Для этого откроем главную страницу личного кабинета. Среди обращений появилось только что отправленное из бота, что видно на рисунке 16.

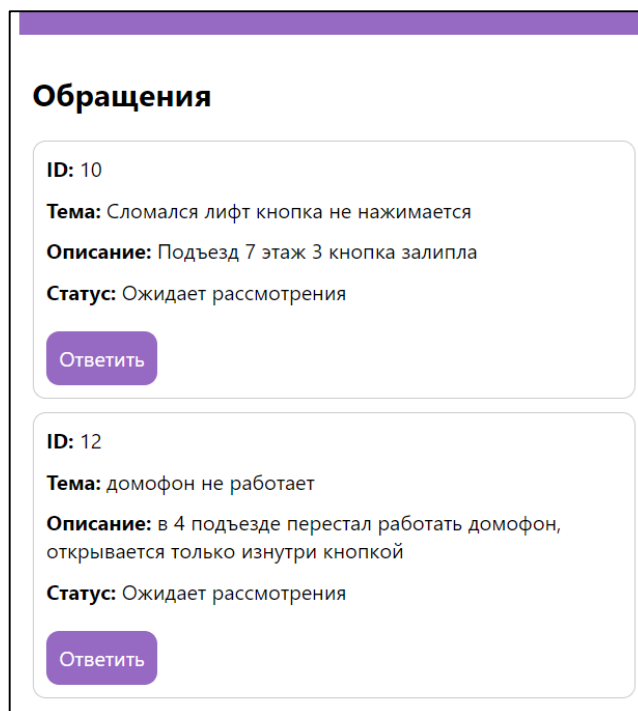


Рисунок 34 – Перечень обращений

Проверим возможность ответить на пришедшее обращение. Для этого необходимо нажать на кнопку «Ответить» и на открывшейся странице ввести ответ на обращение. Страница для ответа на обращение представлена на рисунке 17. После нажатия на кнопку «Отправить» ответ должен быть отправлен пользователю telegram, который отправил данное обращение боту.



### Обращение 12

**Название:** домофон не работает

**Описание:** в 4 подъезде перестал работать домофон, открывается только изнутри кнопкой

**Дата создания:** 2024-05-23T13:22:44

### Ответ на обращение

Введите текст обращения

Рисунок 35 – Ответ на обращение

Для проверки ответа снова откроем диалог с ботом. Как видно по рисунку 18, ответ от председателя доставлен.

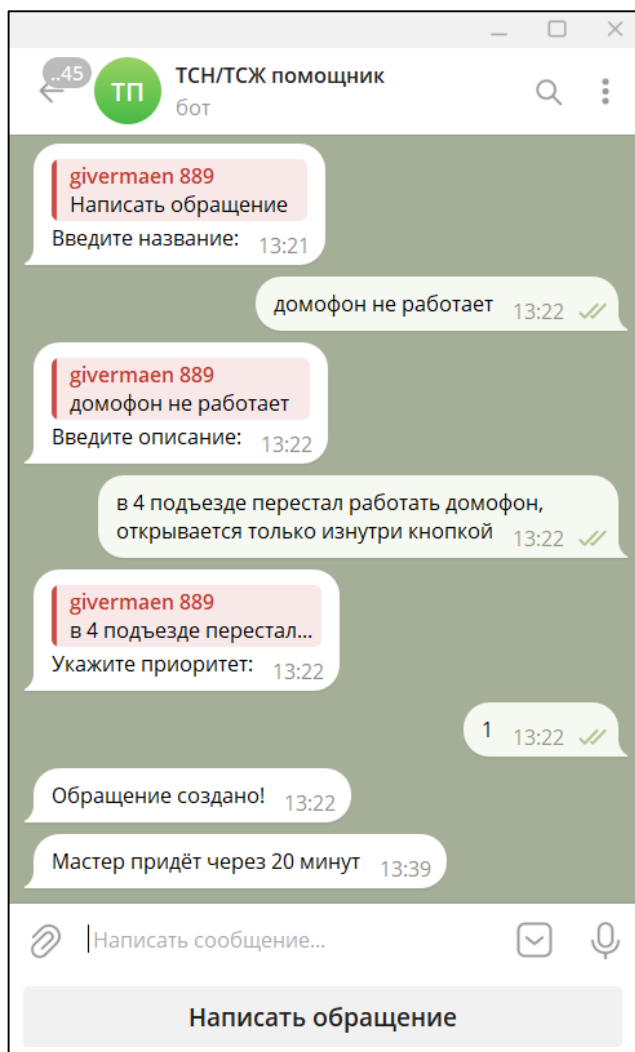


Рисунок 36 – Ответ председателя

Тестирование проведено, систему удовлетворяет поставленным функциональным требованиям. Все основные компоненты системы работают синхронизированно, передача данных между ними проводится без ошибок.

## Заключение

Результатом выпускной квалификационной работы является разработанный прототип информационной системы для управления многоквартирным домом.

Данная работа проводилась в четыре этапа.

На первом этапе был проведён анализ бизнес процессов компании ООО «Квартплата 24» которая занимается разработкой и поддержкой сервисов для упрощения обработки коммунальных платежей для всех участников этого процесса: жителей, управляющих компаний и ресурсоснабжающих организаций. Была рассмотрена структура подразделений компании.

Анализ бизнес процессов показал, что компания имеет потенциал для развития в смежной области деятельности. У компании имеется готовая база клиентов, заинтересованных в использовании отсутствующего на данный момент сервиса.

На втором этапе была рассмотрена предметная область планируемого сервиса. Было обнаружено наличие низкого уровня конкуренции в выбранном направлении. На основе результатов изучения предметной области была описана деятельность потенциальных пользователей без до и после приобретения информационной системы. При сравнении процессов «как есть» и «как должно быть» было определено, что разрабатываемая система обеспечивает значительный уровень автоматизации рутинных действий.

На третьем этапе были детально рассмотрены и смоделированы все процессы, которые должны проходить в разрабатываемой информационной системе. Модели процессов представлены в виде UML диаграмм.

На последнем этапе выполнена разработка и проведены различные виды тестирования готового приложения. Все поставленные цели выполнены.

## Список используемой литературы

1. Шилдт, Герберт. Java. Полное руководство, 12-е изд. : Пер. с англ. – СПб. : ООО «Диалектика», 2023 – 1344с.
2. Вязовик, Н. А. Программирование на Java : курс лекций / Н. А. Вязовик. - Москва : ИНТУИТ, 2016. - 447 с. - Текст : электронный.
3. Уорбэртон, Р. Лямбда-выражения в Java 8. Функциональное программирование — в массы : практическое руководство / Р. Уорбэртон ; пер. с англ. А. А. Слинкина. - 2-е изд. - Москва : ДМК Пресс, 2023. - 194 с.
4. Наир, В. Предметно-ориентированное проектирование в Enterprise Java с помощью Jakarta EE, Eclipse MicroProfile, Spring Boot и программной среды Axon Framework : практическое руководство / В. Наир ; пер. с англ. А. В. Снастина. - Москва : ДМК Пресс, 2020. - 306 с. - ISBN 978-5-97060-872-2.
5. Доуни, А. Алгоритмы и структуры данных. Извлечение информации на языке Java : практическое руководство / А. Доуни. - Санкт-Петербург : Питер, 2018. - 240 с.
6. Васюткина, И. А. Технология разработки объектно-ориентированных программ на JAVA / Васюткина И.А. - Новосибирск :НГТУ, 2012. - 152 с.: ISBN 978-5-7782-1973-1.
7. Уоллс, К. Spring в действии : практическое руководство / К. Уоллс ; пер. с англ. А. Н. Киселева. - 6-е изд. - Москва : ДМК Пресс, 2022. - 544 с. - ISBN 978-5-93700-112-2.
8. Карнелл, Д. Микросервисы Spring в действии : практическое руководство / Д. Карнелл, И. Уайлупо Санчес ; пер. с англ. А. Н. Киселева. - Москва : ДМК Пресс, 2022. - 490 с.
9. Рогов, Е. В. PostgreSQL 15 изнутри / Е. В. Рогов. - Москва : ДМК Пресс, 2023. - 664 с.

10. JavaScript для начинающих / Майк МакГрат ; [перевод с английского М.А. Райтмана]. – 6-е издание. Москва : Эксмо, 2023. – 232 с. : ил.
11. Олег Докука, Игорь Лозинский. Практика реактивного программирования в Spring 5. – М.: ДМК Пресс, 2019. – 508 с.
12. Эванс, Бенджамин Дж., Флэнаган, Дэвид. Java. Справочник разработчика, 7-е изд. : Пер. с англ. – СПб. : ООО «Диалектика», 2019 – 592 с. : ил.
13. Бауэр К., Кинг Г., Грегори Г. Java Persistence API и Hibernate / пер. с англ. Д.А. Зинкевича; под науч. ред. А.Н. Киселёва. – М.: ДМК Пресс, 2017. – 632 с.: ил.
14. Лукас да Коста. Тестирование JavaScript. – СПб.: Питер, 2023. – 592 с.:ил.
15. Васильев А.Н. Java. Объектно-ориентированное программирование: Учебное пособие. – СПб.: Питер, 2013 – 400 с.
16. Блинов, И.Н., Романчик, В.С. Java. Методы программирования : уч.-мет. Пособие /И.Н. Блинов, В.С. Романчик. – Минск : издательство «Четыре четверти», 2013. – 896 с.
17. Буч, Г. Введение в UML от создателей языка : практическое руководство / Г. Буч, Дж. Рамбо, И. Якобсон ; пер. с англ. Н. Мухина. — 3-е изд. - Москва : ДМК Пресс, 2023. - 495 с.
18. Гома, Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений : практическое руководство / Х. Гома ; пер. с англ. А. А. Слинкина. - 2-е изд. - Москва : ДМК Пресс, 2023. - 701 с.
19. Бабич, А. В. UML: Первое знакомство : краткий учебный курс / А. В. Бабич. - Москва : ИНТУИТ, 2016. - 150 с.
20. Буч, Г. Язык UML. Руководство пользователя : практическое руководство / Г. Буч, Дж. Рамбо, И. Якобсон ; пер. с англ. Н. Мухина. — 3-е изд. - Москва : ДМК Пресс, 2022. - 495 с.

21. Леоненков, А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose : краткий курс / А. В. Леоненков. - Москва : ИНТУИТ, 2016. - 140 с.
22. Розенберг, Д. Применение объектного моделирования с использованием UML и анализ прецедентов на примере книжного Internet-магазина : практическое руководство / Д. Розенберг, К. Скотт ; пер. с англ. А. А. Слинкина. - 2-е изд. - Москва : ДМК Пресс, 2022. - 160 с.
23. Ted Hagos. Beginning IntelliJ IDEA: Integrated Development Environment for Java Programming. Apress, 2022 г. – 283 с.
24. OSWALD CAMPESATO. Java basics using ChatGPT / GPT-4. Mercury learning and information, 2024 г. – 709 с.
25. Marten Deinum, Daniel Rubio, Josh Long. Spring 6 Recipes A Problem-Solution Approach to Spring Framework Fifth Edition. Apress, 2023 г. – 774 с.
26. Otávio Santana, Karina Varela. Persistence Best Practices for Java Applications. Packt, 2023 г. – 202 с.
27. Dr. Nirali Dabhi, Dr. Dharmendra Patel, Dr. Atul Patel. Web Development Toolkit for Java Developers. BPB Online, 2023 г. – 518 с.